

## INDEX

JP-A-9-223050



JAPANESE PATENT OFFICE

## PATENT ABSTRACTS OF JAPAN

(11)Publication number: 09223050

(43) Date of publication of application: 26.08.1997

(51)Int.Cl.

G06F 12/00

G06F 12/00

G06F 12/00

**G06F 13/00**

**G06F 13/00**

(21)Application number: 08027577

(71)Applicant:

HITACHI LTD

(22)Date of filing: 15.02.1996

(72)Inventor:

MATSUMOTO KOJI

NAKANISHI HIROSHI

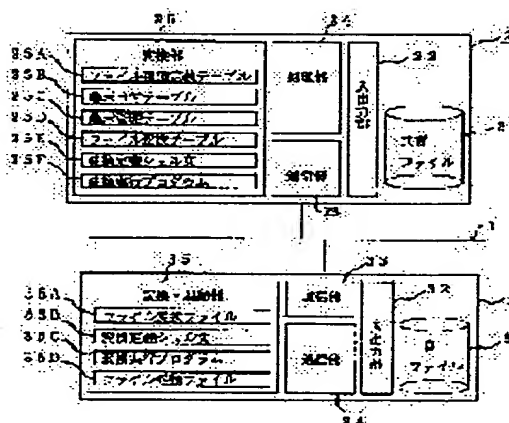
TAKAMURA TOSHIKO

#### (54) USING METHOD FOR SHARED FILE

(57)Abstract:

**PROBLEM TO BE SOLVED:** To use a file of different types of applied software as if it were a file of user's own software with no consciousness of the types of produced applied software by making a server act as a client to perform the processing against the file request given to the server from the client and answering the client about the result after a desired file is acquired.

**SOLUTION:** A file request is given to a server 2 from a client 3. Receiving the request, the server 2 acquires a desired file from a shared file device 21. If the requested file does not exist, the file is acquired from a server 4 of another system. When a file of the server 4 is acquired from the server 2, the server 2 is set as a client of the server 4. In other words, the



server 2 acts as the server of the client 3 and also as the client of the server 4. Thus, the file of the server 4 can be acquired from the server 2.

---

#### LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

---

Copyright (C); 1998 Japanese Patent Office

[MENU](#)

[SEARCH](#)

[INDEX](#)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-223050

(43) 公開日 平成9年(1997)8月26日

(51) Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 12/00	5 1 1		G 0 6 F 12/00	5 1 1
	5 2 0			5 2 0 E
	5 4 5			5 4 5 A
13/00	3 5 1		13/00	3 5 1 B
	3 5 7			3 5 7 Z
審査請求 未請求 請求項の数 2 O L (全 10 頁)				

(21) 出願番号 特願平8-27577

(22) 出願日 平成8年(1996)2月15日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 松本 耕治

神奈川県川崎市幸区鹿島田890番地の12

株式会社日立製作所情報システム事業部内

(72) 発明者 中西 洋

神奈川県川崎市幸区鹿島田890番地の12

株式会社日立製作所情報システム事業部内

(72) 発明者 高村 稔子

茨城県日立市大みか町五丁目2番1号 株

式会社日立製作所大みか工場内

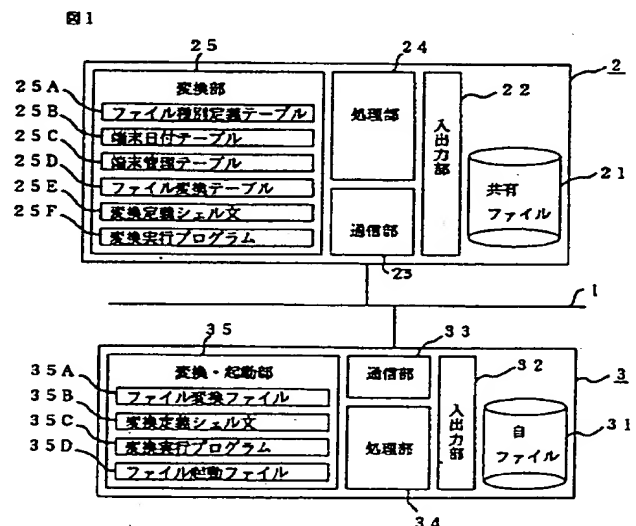
(74) 代理人 弁理士 小川 勝男

(54) 【発明の名称】 共有ファイルの利用方法

(57) 【要約】

【課題】 特定システムで共有化しているファイルについて、その作成された応用ソフトの種別を意識することなく、異なる応用ソフトのファイルを利用可能とする。

【解決手段】 ネットワーク1により接続されているサーバ2とクライアント3から構成されている環境において、サーバ2は共有ファイル装置21と、入出力部22と、通信部23と処理部24と、クライアント3が要求したファイルを利用可能であるかを判別し必要に応じてファイル変換を行う変換部25を備え、クライアント3は自ファイル装置31と、入出力部32と、通信部33と処理部34と、クライアント3が要求したファイルを利用可能であるかを判定し必要に応じてファイル変換およびファイル起動を行う変換・起動部35を備え、クライアント3からのファイル要求の際、作成された応用ソフトの種別を意識することなく、システム側で自動変換・起動させることにより異なる応用ソフトのファイルを利用可能とする。



## 【特許請求の範囲】

【請求項1】 ネットワークにより接続されているサーバとクライアントから構成されている環境において、サーバは共有するファイルを格納する共有ファイル装置と、その共有ファイル装置への入出力を行う入出力部と、クライアントとファイル転送を行うための通信部とファイルの共有に必要なサーバ側処理全般を行う処理部と、クライアントが要求したファイルをクライアントにインストールされている応用ソフトで利用可能であるかを判定し必要に応じてファイル変換を行う変換部を備え、クライアントは独自に所有しているファイルを格納する自ファイル装置と、その自ファイル装置への入出力を行う入出力部と、サーバとファイル転送を行うための通信部とファイルの共有に必要なクライアント側処理全般を行う処理部と、クライアントが要求したファイルをクライアントにインストールされている応用ソフトでどのように加工すれば利用可能であるかを判定し必要に応じてファイル変換およびファイル起動を行う変換・起動部を備え、クライアントからのファイル要求の際、作成された応用ソフトの種別を意識することなく、システム側で自動変換・起動させることにより異なる応用ソフトのファイルを利用可能とすることを特徴とする共有ファイルの利用方法。

【請求項2】 ネットワーク1により接続されているサーバ2とクライアント3、サーバ4、クライアント5から構成されている環境において、サーバ2とクライアント3は同じグループに属するシステムとして存在し、サーバ4とクライアント5はそれとは別の同じグループに属するシステムとして存在し、クライアント3が別のグループのサーバ4のファイルにアクセスする場合において、クライアント3からサーバ4へのファイル要求に対して、サーバ4のクライアントとなったサーバ2がその処理を代行し、目的のファイルを取得後、その結果をクライアント3に回答することを特徴とする複数システムおよび複数サーバ間における共有ファイルの利用方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は共有ファイルの利用方法に係り、特にクライアントサーバ型のシステムにおいて共有化しているファイルを、その作成された応用ソフトの種別を意識することなく、異なる応用ソフトのファイルを利用可能とする共有ファイルの利用方法に関する。

## 【0002】

【従来の技術】 従来、共有ファイルの利用方法について、例えば、大型計算機を使用したシステムにおいては次のようなファイルの利用方法が確立されている。これは、大型計算機に接続した異なる種類の端末で大型計算機のファイルを利用するために、それぞれの端末の環境を大型計算機で管理し、端末からのファイル要求の際に

その端末の環境に合わせて大型計算機上でファイルデータ変換して転送するという仕組みである。

【0003】 一方、中小計算機を相互接続したクライアント／サーバ環境においては大型計算機を使用したシステムとは異なり、独立した計算機間で作成された様々な応用ソフトのファイルを任意の計算機上で共有する仕組みが必要である。そのため様々なファイル共有方式が考案されている。例えば、当初から考案されているOS Operating System、NOS Network Operating Systemなど基本ソフトの機能で実現しているファイル共有方式がある。さらに近年では、計算機の利用による複数作業者の同時協調的な作業支援を目的としたグループウェア、PDM Product Data Managementなどを用いて、利用者の操作性を向上したファイル共有方式がある。これらファイル共有方式も近年の利用者数の増大や、管理ファイルの複雑化などに伴って、単一サーバだけでなく、複数サーバに跨ってファイル共有する仕組みが確立されている。このように様々なファイル共有方法が確立されているが、そのファイルの利用方法については一般に、クライアントからネットワークで接続されたサーバの共有ファイルを指示することにより、サーバからクライアントにファイル転送され、クライアントで利用可能とするという仕組みである。

## 【0004】

【発明が解決しようとする課題】 上記、従来の技術においては、クライアント／サーバ環境で様々なファイル共有方式が確立されているが、その利用においては次の問題点がある。第1に、ある応用ソフトで作成されたファイルの再利用は、必ず作成した応用ソフトを用いて利用しなければならない点である。実際にファイル共有の運用を開始するとよほど運用を考えていない限り、利用者毎に使用する応用ソフトが異なり、異なる応用ソフト間のファイル共有が不可能となってしまう。運用で対応する場合には、特定のサーバに接続する複数のクライアントにおいて、応用ソフト種別毎に特定の応用ソフトを使用するなど応用ソフトを規定し一致させる必要があった。しかし、情報化社会の浸透に伴い、共有ファイルの利用範囲が単一サーバから複数サーバに、一部門だけではなく複数部門に、さらには一企業だけでなく複数企業に跨って広がる傾向にあり、前述した運用で対応する方法にも限界が見えてきた。

【0005】 第2に、ファイル共有を実現する2つ以上の異なるシステムが存在する場合、お互いのファイル共有が困難である点である。前述したように共有ファイルの利用範囲が拡大しており、ファイル共有を実現する2つ以上の異なるシステムが必ず存在するようになることである。現状、異なるファイル共有を実現するシステムの共有ファイルを利用する場合、そのクライアントが所有する現在のクライアント環境に、その異なるシステムのサーバと通信可能とするもう1つのクライアント環境を

追加しなければならない。異なるシステムのサーバに対する参照頻度は低いにもかかわらず、その他の全クライアントが異なるシステムのサーバと通信できるよう設定しなければならない。これではクライアント毎の設定が複雑になるだけでなくクライアント当りの投資も高くなってしまう。さらに異なるシステムの種類が増えれば増えるほど、その問題点は顕著となる。

【0006】本発明の目的は、第1に特定システムで共有化しているファイルについて、その作成された応用ソフトの種別を意識することなく、異なる応用ソフトのファイルをあたかも自分の応用ソフトで作成したファイルとして利用可能とする手段を提供することにある。

【0007】本発明の第二目的は、ファイル共有を実現する2つ以上の異なるシステムで管理されている共有ファイルを容易に利用可能とする手段を提供することにある。

【0008】

【課題を解決するための手段】上記、本発明の目的を達成するため、本発明の第1の特徴は、ネットワーク1により接続されているサーバ2とクライアント3から構成されている環境において、サーバ2は共有するファイルを格納する共有ファイル装置21と、その共有ファイル装置への入出力を行う入出力部22と、クライアント3とファイル転送を行うための通信部23とファイルの共有に必要なサーバ側処理全般を行う処理部24と、クライアント3が要求したファイルをクライアント3にインストールされている応用ソフトで利用可能であるかを判別し必要に応じてファイル変換を行う変換部25を備え、クライアント3は独自に所有しているファイルを格納する自ファイル装置31と、その自ファイル装置への入出力を行う入出力部32と、サーバ2とファイル転送を行うための通信部33とファイルの共有に必要なクライアント側処理全般を行う処理部34と、クライアント3が要求したファイルをクライアント3にインストールされている応用ソフトでどのように加工すれば利用可能であるかを判定し必要に応じてファイル変換およびファイル起動を行う変換・起動部35を備え、クライアント3からのファイル要求の際、作成された応用ソフトの種別を意識することなく、システム側で自動変換・起動させることにより異なる応用ソフトのファイルを利用可能とすることである。

【0009】本発明の第2の特徴は、ネットワーク1により接続されているサーバ2とクライアント3、サーバ4、クライアント5から構成されている環境において、サーバ2とクライアント3は同じグループに属するシステムとして存在し、サーバ4とクライアント5はそれとは別の同じグループに属するシステムとして存在し、クライアント3が別のグループのサーバ4のファイルにアクセスする場合において、クライアント3からサーバ4へのファイル要求に対して、サーバ4のクライアントと

なったサーバ2がその処理を代行し、目的のファイルを取得後、その結果をクライアント3に回答することである。

【0010】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照し説明する。

【0011】

【実施例1】まず、図1は本実施例1の全体構成を示すブロック図である。サーバ2とクライアント3はネットワーク1により接続されている。サーバ2は共有するファイルを格納する共有ファイル装置21と、その共有ファイル装置への入出力を行う入出力部22と、クライアント3とファイル転送を行うための通信部23とファイルの共有に必要なサーバ側処理全般を行う処理部24と、クライアント3が要求したファイルをクライアント3にインストールされている応用ソフトで利用可能であるかを判定し必要に応じてファイル変換を行う変換部25より構成されている。クライアント3は独自に所有しているファイルを格納する自ファイル装置31と、その自ファイル装置への入出力を行う入出力部32と、サーバ2とファイル転送を行うための通信部33とファイルの共有に必要なクライアント側処理全般を行う処理部34と、クライアント3が要求したファイルをクライアント3にインストールされている応用ソフトでどのように加工すれば利用可能であるかを判定し必要に応じてファイル変換およびファイル起動を行う変換・起動部35より構成されている。次に、本発明の骨子となる変換部25と変換・起動部35の詳細について示す。

【0012】最初に、変換部25について示す。変換部25はファイル種別定義テーブル25Aと、端末日付テーブル25Bと、端末管理テーブル25Cと、ファイル変換テーブル25Dと、変換定義シェルスクリプト25Eと、変換実行プログラム25Fの6ブロックから構成されている。上記テーブルは複数のクライアントで共有するため、更新時には排他制御を行い整合性を確保する。

【0013】次に、変換部25のファイル種別定義テーブル25Aに関する詳細図を図2に示す。ファイル種別定義テーブル25Aとは共有するファイルがどの応用ソフトで扱えるファイルであるかを定義するテーブルである。共有するファイルのファイル形式やファイル拡張子などの特徴から、どの応用ソフト種別のどの応用ソフトのファイルであるかを定義する。このファイルの構成項目としては、応用ソフト種別201、応用ソフト名称202、応用ソフト判定文203が必要となる。例えば、応用ソフト種別201がワードプロセッサ、応用ソフトワードプロセッサ1aの応用ソフト名称202を「ワープロ1a」とし、このファイルが記憶装置に格納される際に付されるファイル名のファイル拡張子「.doc」で識別できる場合、図2の◆で示すように定義する。

【0014】次に、変換部25の端末日付テーブル25

Bに関する詳細図を図3に示す。端末日付テーブル25Bとはファイルを共有する全てのクライアント3について、ファイル共有に必要なクライアント3の環境を定義している環境設定ファイル後述するファイル変換ファイル35Aとファイル起動ファイル35Dの2ファイルを指すの最新ファイル日付を自動的に保持するテーブルである。これは、クライアント3の環境に変更があるかを監視し、変更があった場合はクライアント3側の設定内容を元にサーバ2側を自動的に更新する判断の役割を担うものである。このテーブルの属性項目としては、端末識別名称301、ファイル変換ファイル最新日付302、ファイル起動ファイル最新日付303が必要となる。具体的には、端末識別名称は、ホスト名など端末を特定できる名称を定義する。ファイル変換ファイル最新日付は「1995.01.01」、ファイル起動ファイル最新日付は「1995.02.02」の場合、図3の◆で示すように定義する。

【0015】次に、変換部25の端末管理テーブル25Cに関する詳細図を図4に示す。端末管理テーブル25Bとはファイルを共有する全てのクライアント3について、それらにインストールされている応用ソフトのうち、各クライアント3で利用可能な応用ソフトが定義され、さらにその中で応用ソフト種別毎の代表となる応用ソフトが1つ定義されているテーブルである。このテーブルは各クライアント3の環境設定ファイルを元にして自動生成される。このテーブルの属性項目としては、端末識別名称401、応用ソフト名称402、代表フラグ403が必要となる。端末識別名称は、前述したようにホスト名など端末を特定できる名称を定義する。応用ソフト名称は、ファイル種別定義テーブル25Aで定義したものを定義する。代表フラグは、応用ソフト種別毎に代表となるものを「1」とし、それ以外のものを「0」と定義する。端末識別名称が「PC001」、応用ソフト名称が「ワープロ1a」、これが応用ソフト種別で代表となる応用ソフトである場合、図4の◆で示すように定義する。

【0016】次に、変換部25のファイル変換テーブル25Dに関する詳細図を図5に示す。ファイル変換テーブル25Dとは応用ソフト種別毎に、応用ソフト間のファイル変換を実行するためのコマンドを定義するテーブルである。このテーブルの属性項目としては、変換元応用ソフト名称501、変換先応用ソフト名称502、ファイル変換実行コマンド503で構成される。例えば、応用ソフト名称「ワープロ1a」のファイル「ファイル1a」を、応用ソフト名称「ワープロ2」のファイル「ファイル2」に変換するファイル変換実行コマンドを「ワープロ1a2」とした場合、図5の◆で示すように定義する。

【0017】次に、変換部25の変換定義シェル文25Eに関する詳細図を図6に示す。変換定義シェル文25

Eとはファイル変換テーブル25Dに記述するファイル変換実行コマンドを1コマンドで記述できない場合、これを補うために定義するものである。応用ソフトのファイル変換をダイレクトに変換するプログラムがある場合には使用しなくてよい。但し、応用ソフト間でダイレクト変換するプログラムは希有であり、CAL S Continuous Acquisition and Lifecycle Supportなどで唱えられているデータ交換のための標準データ形式のファイルを経由し変換する方法が現実的となる。そのため、本変換定義シェル文が必要となる。例えば、応用ソフト名称「ワープロ1a」のファイル「ファイル1a」を応用ソフト名称「ワープロ2」のファイル「ファイル2」に変換するファイル変換実行コマンドシェル文を「ワープロ1a2」とし、直接変換が不可能であるため、一旦、標準データのファイル「ファイル標準」を仲介させ、2コマンドで変換処理を実現させる場合、図6の◆で示すように定義する。

【0018】次に、変換部25の変換実行プログラム25Fとは元のファイル形式とは異なるファイルへ実際にファイル変換を行うためのプログラムである。これは、応用ソフト開発元などから提供されるプログラムである。

【0019】続いて、変換・起動部35について示す。変換・起動部35はファイル変換ファイル35Aと、変換定義シェル文35Bと、変換実行プログラム35Cと、ファイル起動ファイル35Dの4ブロックから構成されている。

【0020】まず、変換部・起動部35のファイル変換ファイル35Aに関する詳細図を図7に示す。ファイル変換ファイル35Aとは前述したファイル変換テーブル25Dと同様に応用ソフト種別毎に応用ソフト間のファイル変換を実行するためのコマンドを定義するファイルである。このファイルはサーバ2とは関係なくクライアント3の環境にクローズした世界で定義する必要がある。このファイルの属性項目としては、ファイル変換テーブル25Dと同様なので省略する。例えば、応用ソフト名称「ワープロ3」のファイル「ファイル3」を、応用ソフト名称「ワープロ2」のファイル「ファイル2」に変換するファイル変換実行コマンドを「ワープロ32」とした場合、図7の◆で示すように定義する。このことにより「ワープロ3」の応用ソフトがない場合でも「ワープロ2」の「ファイル2」を取得できるので利用可能となる。

【0021】次に、変換部・起動部35の変換定義シェル文35Bについて示す。変換定義シェル文35Bは変換定義シェル文25Eと記述方法については同様となるためここでは省略する。但し、このシェル文はサーバ2とは関係なくクライアント3の環境にクローズした世界で定義する必要がある。

【0022】第3に、変換部・起動部35の変換実行プ

ログラム35Cとは変換実行プログラム25Fと同様に元のファイル形式とは異なるファイルへ実際にファイル変換を行うためのプログラムである。但し、このプログラムもサーバ2とは関係なくクライアント3の環境にクローズした世界で定義する必要がある。

【0023】第4に、変換部・起動部35のファイル起動ファイル35Dに関する詳細図を図8に示す。ファイル起動ファイル35Dとは応用ソフト毎に任意のファイルを読み込み応用ソフトを起動するための起動実行コマンドを定義するファイルである。また応用ソフト種別毎に代表する応用ソフトを1つ選定し定義しておく。このファイルの属性項目としては、代表フラグ、応用ソフト名称、ファイル起動実行コマンドで構成される。代表フラグ、応用ソフト名称については端末管理テーブル25Cと同様に定義する。ファイル起動実行コマンドは、起動コマンド名称、オプション、読み込むファイル名称などで定義する。ファイルの種別が異なってもオプション起動の使い分けでファイルを利用できる応用ソフトの場合に有効となる。

【0024】クライアント3の環境ファイルであるファイル変換ファイル35Aとファイル起動ファイル35Dは必要に応じて任意に作成、更新、削除が可能である。これら変更のサーバ2側への反映はファイル日付の比較により自動的に行われる。つまり、クライアント3側のファイル日付の方がサーバ2側のそれと比較し最新日付である場合、自動更新処理を行う。次に、クライアント3からサーバ2に対してファイル要求を行う場合の処理の流れについて図9～10を参照し説明する。まず、クライアント3からサーバ2に対してファイル要求を行う(ステップ901)。ファイル要求時に必要な情報は、要求ファイル名称、端末識別名称、ファイル変換ファイル35Aとファイル起動ファイル35Dのファイル日付である。サーバ2はクライアント3のファイル要求を受けて(ステップ902)、端末識別名称をもとにサーバ2の端末日付テーブル25Bとマッチングを行い、ファイル日付の比較を行う(ステップ903)。端末日付テーブル25Bの最新日付と要求時のファイル日付が一致している場合、クライアント3の環境に変更がないと判断する。逆に最新日付より要求時のファイル日付が最新である場合、端末日付テーブル25Bおよび端末管理テーブル25Cの更新処理を行う。更新処理は、まずクライアント3に変更のある環境ファイルを要求し(ステップ904)、その記述内容にエラーがないかどうかをチェックし(ステップ905)、エラーがない場合は排他処理を行い、端末日付テーブル25B、端末管理テーブル25Cを更新する(ステップ906)。入力エラーがある場合は更新処理は行わずエラー表示を行い終了する。次に、目的の要求ファイルを共有ファイル装置21より取得する。この際要求ファイルが存在しない場合はエラー表示を行い終了する(ステップ907)。要求

ファイルが存在する場合、次に要求ファイルの応用ソフト種別と応用ソフト名称について、ファイル種別定義テーブル25Aと照合し決定する(ステップ908)。この際ファイル種別定義テーブル25Aのファイルとして該当しない場合はエラー表示を行い終了する。要求ファイルの応用ソフト種別、応用ソフト名称が決定した場合、クライアント3がこのファイルを利用可能であるか、端末管理テーブル25Cの端末識別名称をキーとして照合して判定する(ステップ909)。照合した結果、要求ファイルの応用ソフト名称が端末管理テーブル25Cに定義されていた場合、このファイルはクライアント3で処理可能と判断される。照合の結果、定義されていなかった場合、このままのファイルではクライアント3で処理が不可能のため、サーバ2でクライアント3の代表応用ソフトにファイル変換可能であるか判定する(ステップ910、911)。判定方法は要求ファイルの応用ソフト種別をもとに端末管理テーブル25Cの代表フラグと照合し、クライアント3で処理できる応用ソフト名称が決定できる。この応用ソフト名称をキーとして、ファイル変換テーブル25Dを検索し、該当するファイル変換実行コマンドが存在する場合、記述されたコマンド文に従って変換実行プログラム25Fを実行しファイル変換を行う(ステップ912)。この際定義内容に応じて変換定義シェル文25Eを参照し、変換実行プログラム25Eを実行する場合もある。逆に該当しない場合は、エラー表示を行い終了する(ステップ919)。要求ファイルがクライアント3で処理可能となった場合、そのファイルをサーバ2からクライアント3にネットワーク1経由でファイル送信する(ステップ913、914)。クライアント3は受信したファイルについてそのまま起動可能か、あるいは一旦ファイル変換する必要があるかを判定する(ステップ915)。そのまま起動できるかを判定するには、ファイルの応用ソフト名称がファイル起動ファイル35Dのなかに定義されている場合、そのまま起動できる(ステップ916)。逆に定義されていない場合は、ファイル変換ファイル35Aのファイル変換実行コマンドを実行しファイル変換した後、ファイル起動を行う(ステップ917)。ファイル変換はサーバ側と同様に必要に応じて変換定義シェル文35Bを呼び出し、変換実行プログラム35Cで行う。ファイルの起動はファイル起動ファイル35Dよりその方法が決定され、ファイル起動実行コマンドを実行して行う(ステップ918)。

#### 【0025】

【実施例2】次に本実施例2について図面を参照し説明する。まず、図11が本実施例2の全体構成を示すブロック図である。実施例2では、複数の異なるサーバ間でファイル共有を実現する場合についての適用を示す。サーバ2、クライアント3、サーバ4、クライアント5はネットワーク1により接続されている。このサーバ2、

クライアント3は実施例1と同じものであり、新たにサーバ4とクライアント5を加えた構成が実施例2の構成である。サーバ4はサーバ2とクライアント5はクライアント3と同じ構成である。これらの関係は、サーバ2とクライアント3は同じグループに属するシステムとし、サーバ4とクライアント5はそれとは別の同じグループに属するシステムとする。

【0026】次にクライアント3からのファイル要求に対して、要求ファイルの格納先が別システムのサーバ4に存在する場合の処理の流れについて図12～13を参照し説明する。本フローチャートにおいて、基本的には図9と図10の流れと同じであり、ステップ901～919はステップ1201～1219に相当し（下2桁を合わせてある）、新たに加わったのはステップ1220、1221である。

【0027】まず、クライアント3からサーバ2に対してファイル要求を行う（ステップ1201）。サーバ2はクライアント3のファイル要求を受けて目的の要求ファイルを共有ファイル装置21より取得する（ステップ1202）。この際要求ファイルが存在しない場合は、別システムのサーバ4より取得する（ステップ1207、1220）。さらに要求ファイルが存在しない場合はエラー表示を行い終了する（ステップ1221）。

【0028】要求ファイルがサーバ2に存在する場合、以下実施例1と同じ流れに従って処理される（ステップ1208～）。要求ファイルが別のサーバ4に存在する場合、サーバ2とサーバ4間のファイル授受が発生する。ここで、そのサーバ2とサーバ4におけるファイル授受の仕組みについて説明する。サーバ4のファイルをサーバ2から取得する場合、サーバ2はサーバ4のクライアントとして設定しておく。つまり、サーバ2はクライアント3のサーバとしての役割と、サーバ4のクライアントとしての役割をもつ。このことにより、サーバ4にあるファイルをサーバ2から取得することができる（ステップ1221）。要求ファイルを取得後、以下実施例1と同じ流れに従って処理される（ステップ1208～）。

【0029】

【発明の効果】本発明を用いると、第1に特定システムで共有化しているファイルについて、その作成された応用ソフトの種別を意識することなく、異なる応用ソフトのファイルをあたかも自分の応用ソフトで作成したファイルとして利用可能となる。第2にファイル共有を実現する2つ以上の異なるシステムで管理されている共有ファイルを容易に利用可能となる。

【図面の簡単な説明】

【図1】 本発明の実施例1の全体構成を示すブロック

図である。

【図2】 変換部のファイル種別定義テーブルに関する詳細図である。

【図3】 変換部の端末日付テーブルに関する詳細図である。

【図4】 変換部の端末管理テーブルに関する詳細図である。

【図5】 変換部のファイル変換テーブルに関する詳細図である。

【図6】 変換部の変換定義シェル文に関する詳細図である。

【図7】 変換・起動部のファイル変換ファイルに関する詳細図である。

【図8】 変換・起動部のファイル起動ファイルに関する詳細図である。

【図9】 実施例1の処理の流れを示すフローチャートである（その1）。

【図10】 実施例1の処理の流れを示すフローチャートである（その2）。

【図11】 本発明の実施例2の全体構成を示すブロック図である。

【図12】 実施例2の処理の流れを示すフローチャートである（その1）。

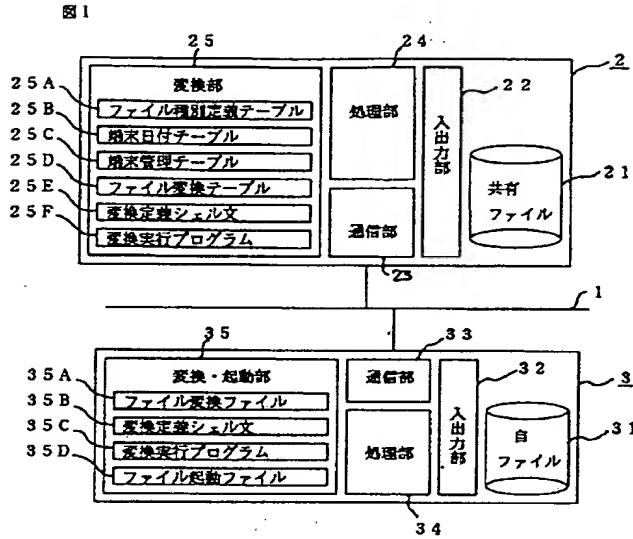
【図13】 実施例2の処理の流れを示すフローチャートである（その2）。

【符号の説明】

1	;	ネットワーク			
2	;	サーバ	3	;	クライアント
21	;	共有ファイル装置	31	;	自ファイル装置
22	;	入出力部	32	;	入出力部
23	;	通信部	33	;	通信部
24	;	処理部	34	;	処理部
25	;	変換部	35	;	変換・起動部
25A	;	ファイル種別定義テーブル	35A	;	ファイル変換ファイル
25B	;	端末日付テーブル	35B	;	変換定義シェル文
25C	;	端末管理テーブル	35C	;	変換実行プログラム
25D	;	ファイル変換テーブル	35D	;	ファイル起動ファイル
25E	;	変換定義シェル文			
25F	;	変換実行プログラム			
4	;	サーバ	5	;	クライアント



【図1】



【図2】

図2  
ファイル種別定義テーブル25A

201 応用ソフト種別	202 応用ソフト名称	203 応用ソフト判定文
◆ ワードプロセッサ	ワープロ1a	拡張子 = "¥. doc"
ワードプロセッサ	ワープロ1b	拡張子 = "¥. dot"
ワードプロセッサ	ワープロ2	拡張子 = "¥. xxx"
:	:	:

【図3】

図3  
端末日付テーブル25B

301 端末識別名称	302 ファイル変換ファイル 最新日付	303 ファイル起動ファイル 最新日付
◆ PC001	1995. 01. 01	1995. 02. 02
PC002	1995. 01. 12	1995. 02. 10
PC003	1995. 01. 04	1995. 01. 04
PC004	1995. 01. 20	1995. 01. 25
:	:	:

【図4】

図4  
端末管理テーブル

401 端末識別名称	402 応用ソフト名称	403 代表フラグ
◆ PC001	ワープロ1a	1
	ワープロ1b	0
	ワープロ2	0
	ワープロ3	0
	ワープロ4	0
	表計算1	1
	表計算2	0
	表計算3	0
PC002	ワープロ2	1
	CAD1	0
	CAD2	1
:	:	:

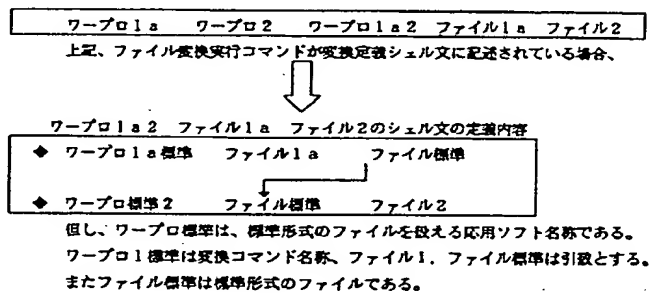
【図5】

図5  
ファイル変換テーブル

501 変換元 応用ソフト名称	502 変換先 応用ソフト名称	503 ファイル変換実行コマンド
◆ ワープロ1a	ワープロ2	ワープロ1a2 ファイル1a ファイル2
ワープロ1b	ワープロ3	ワープロ1b3 ファイル1b ファイル3
ワープロ2	ワープロ1a	ワープロ21a ファイル2 ファイル1a
ワープロ3	ワープロ1b	ワープロ31b ファイル3 ファイル1b
表計算1	表計算2	表計算12 ファイル1 ファイル2
:	:	:

【図6】

図6  
変換定義シェル文



【図 7】

図 7 ファイル変換ファイル

#	変換元	変換先	ファイル変換実行コマンド	
#	応用ソフト名称、応用ソフト名称			
◆	ワープロ3	ワープロ2	ワープロ3 2	ファイル3 ファイル2
	ワープロ4	ワープロ1a	ワープロ4 1a	ファイル4 ファイル1a
	表計算3	表計算1	表計算3 1	ファイル3 ファイル1
	:	:	:	:

但し、#は、その文をコメント文とする

【図 8】

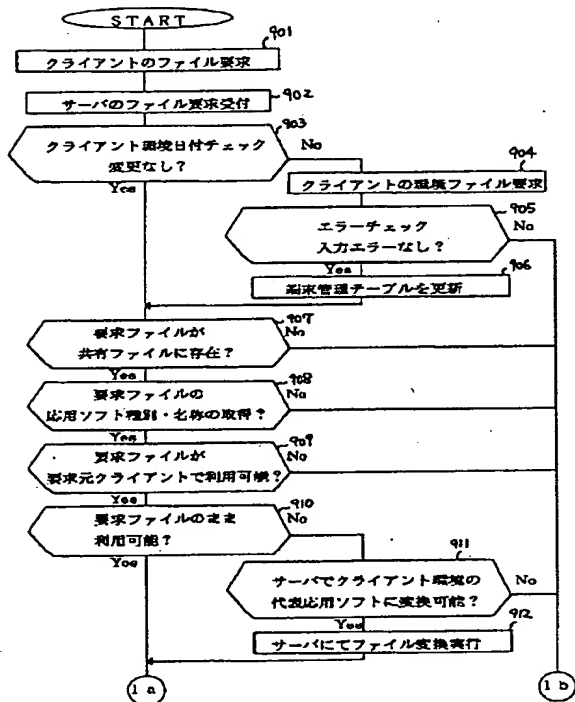
図 8 ファイル起動ファイル

#	代表	応用ソフト名称	ファイル起動実行コマンド	
#	フラグ			
◆	1	ワープロ1a	ワープロ1 -a	ファイル1a
	0	ワープロ1b	ワープロ1 -b	ファイル1b
	0	ワープロ2	ワープロ2	ファイル2
	1	表計算1	表計算1	ファイル1
	0	表計算2	表計算2	ファイル2
	:	:	:	:

但し、#は、その文をコメント文とする

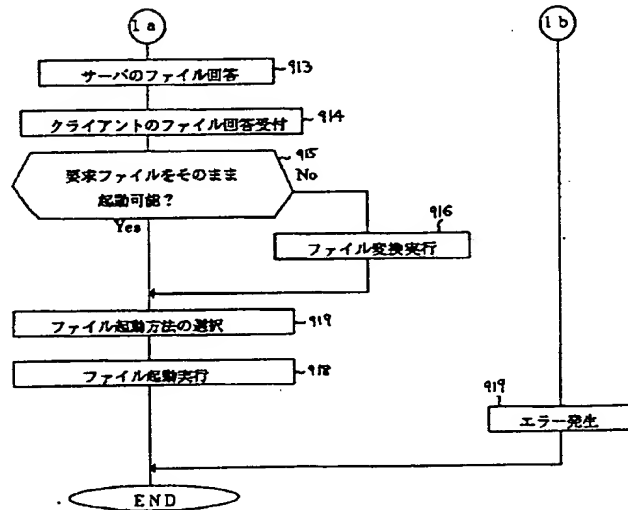
【図 9】

図 9



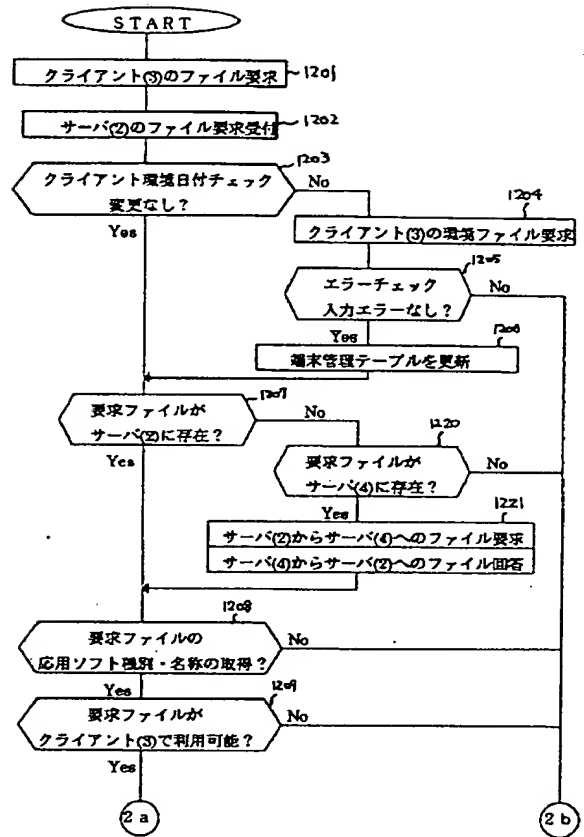
【図 10】

図 10



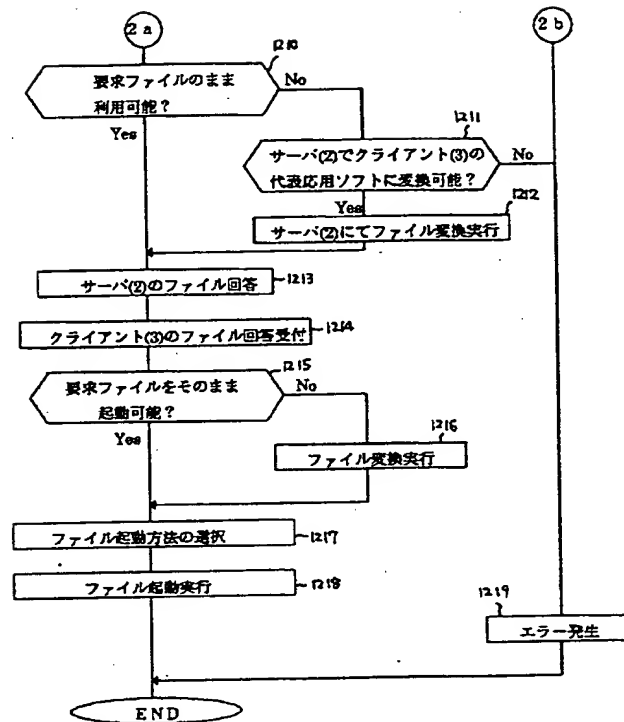
【图 12】

**12**



【図13】

図13



出力ノードに要求データ出力処理手順を登録する（ステップ22138）。最後に、分解結果に対して評価がすべて終了すれば（ステップ22139）、終了する。

【0056】図13において、コード生成222は、処理手順候補が唯一か否かをチェックする（ステップ2220）。唯一であれば、ステップ2223へ移行する。唯一でなければ、カラム値分布情報等からなる最適化情報を処理手順に埋込み（ステップ2221）、問合せ実行時に代入された定数に基づいて処理手順を選択するデータ構造を作成する（ステップ2222）。最後に、処理手順を実行形式へ展開する（ステップ2223）。

【0057】図14において、動的最適化処理223は、作成されている処理手順が単一か否かをチェックする（ステップ22300）。単一であれば、終了する。単一でなければ、代入された定数を基に選択率を算出する（ステップ22301）。処理手順候補に並列な処理手順が含まれるか否かをチェックする（ステップ22302）。含まれていなければ、アクセスパスの選択基準に従って処理手順を選択し（ステップ22313）、終了する。含まれていれば、ディクショナリから最適化情報（結合カラムのカラム値分布情報、アクセス対象となる表のロウ数、ページ数等）を入力し（ステップ22303）、データ取り出し／データ分配のための処理時間を各システム特性を考慮し、前述したように算出する（ステップ22304）。当処理時間から結合処理に割当てるノード数 $p$ を決定し、当処理手順 $a1$ を決定する（ステップ22305）。データ取り出し／データ分配処理時間にバラツキがあるか否かをチェックする（ステップ22306）。バラツキがあれば、データ取り出し／データ分配処理ノード群でスロットソート処理を実行する処理手順 $a2$ を設定する（ステップ22307）。つぎに、結合ノード割当て数 $p$ を $\alpha$ 台だけ増した処理手順 $a3$ を設定する（ステップ22308）。要求データ処理時間が突き合わせ処理時間と1回分のNウェイマージ処理時間との和より大であれば（ステップ22309）、突き合わせ処理へ1回分のNウェイマージ処理を移した処理手順 $a4$ を設定する（ステップ22310）。処理手順 $a1 \sim a4$ で最適な処理手順を応答時間最小、各ノード負荷量最小、他トランザクション応答性能への影響小等の観点で選択する（ステップ22311）。データ分配情報を最適化情報を基にして作成する（ステップ22312）。最適化情報がなければ、ハッシュ関数の結合カラム評価値に従い、データ分配情報を作成する。アクセスパスの選択基準に従って処理手順を選択し（ステップ22313）、終了する。

【0058】図15において、コード解釈実行処理224では、設定された各ノードにおいてそれぞれ対応する処理手順にしたがって処理を行う。

【0059】まず、各ノードでは、データ取り出し／データ分配処理が設定されているか否かを判断する（ス

テップ22400）。データ取り出し／データ分配処理が設定されていれば、各ノードの記憶装置に格納されているデータベースにアクセスし、条件式を評価する（ステップ22401）。最適化情報を基に作成されたデータ分配情報に基づいて、データを取り出し、各結合ノードのバッファへ逐次データを分配する（ステップ22402）。各結合ノードのバッファが満杯か否かを判定し、満杯であれば、ページ形式で対応する結合ノードへ転送する。問い合わせに対応する全てのデータを取り出して分配すると処理が終了する（ステップ22404）。また、各ノードでは、スロットソート処理が設定されているか否かを判断する（ステップ22405）。スロットソート処理が設定されていれば、前記データ取り出し／データ分配処理ノードからのページ形式のデータを受信し、（ステップ22406）受信したデータについて順次スロットソート処理を行う（ステップ22407）。処理したスロットソート結果を一時保存しておき、スロットソート処理を終了する（ステップ22408）。また、Nウェイマージ処理が設定されているか否かを判断する（ステップ22409）。Nウェイマージ処理が設定されていれば、スロットソート結果に基づいてNウェイマージ処理を実行し（ステップ22410）、Nウェイマージ処理結果をバッファなどに一時保存し（ステップ22411）、Nウェイマージ処理を終了する。また、突き合わせ処理が設定されているか否かを判断する（ステップ22412）。突き合わせ処理が設定されていれば、Nウェイマージ処理結果のソートリストを突き合わせ、出力用バッファにデータを設定する（ステップ22413）。出力用バッファが満杯の場合には、ページ形式で要求データ出力ノードへ転送する（ステップ22415）。また、要求データ出力処理が設定されているか否かを判断する（ステップ22416）。要求データ出力処理が設定されていれば、結合ノードからページ形式のデータの転送があるかないかを判断する（ステップ22417）。ページ形式のデータの転送がある場合には、該ページ形式のデータを受信し（ステップ22418）、アプリケーションプログラムへ問い合わせ処理結果を出力し、ページ形式のデータ転送がない場合にはそのまま問い合わせ処理結果を出力する（ステップ22419）。

【0060】また、上記コード解釈実行処理において、処理時間にバラツキがある場合などのときにデータ取り出し／データ分配処理ノード群でスロットソート処理を実行する場合には、データ取り出し／データ分配処理終了後、コード解釈実行処理224を再度実行し、スロットソート処理を行うようにする。

【0061】さらに、ステップ22413でNウェイマージ処理結果が完全ソート列でなければ、最終段のマージと突き合わせ処理とを行う。

【0062】以上のように処理することによりデータベ

ース管理システムの問い合わせ応答時間を短縮することができる。

【0063】図6に示す結合ノード割当て方法と、図7、図8および図9に示すチューニング方法とは、各々独立に適用してもよいし、また任意の組合せで適用してもよい。すなわち、動的最適化処理223では、すべての組合せが適用できる場合を想定している。さらに、データ取り出し処理においては、複数ディスク装置からなる並列入出力アクセス方法の適用と、一括入出力方法／先読み入出力方法の適用と、データ分配処理に最適化情報あるいはハッシュ関数によるデータ分配方法の適用と、Nウェイマージ処理に並列ソート方法の適用と、突き合わせ処理にノード間での突き合わせ処理方法の適用と、要求データ出力処理に複数のノードを割当て並列受け取り処理方法の適用等も考えられる。上記ステップ22309およびステップ22310では、1回分のNウェイマージ処理を仮定しているが、一般的にn回 ( $n \geq 1$ ) としてもよい。

【0064】図4に示す並列パイプライン動作に関して、上記図6に示した結合ノード割当て方法と、図7、図8および図9に示すチューニング方法とを適用すると、取り出しフェーズ、マージフェーズ、結合フェーズの内、マージフェーズが省略可能となる場合も存在する。すなわち、スロットソート連長の延び、Nウェイマージ処理の移動により可能となる。この場合、問合せ実行処理でもマージフェーズの処理を省略する。

【0065】本発明の問合せ処理方法は、統計情報を用いた規則とコスト評価との併用に限らず、適当なデータベース参照特性情報を与える処理手順が得られるものであれば適用できる。例えば、コスト評価のみ、規則利用のみ、コスト評価と規則利用の併用等の最適化処理を行うDBMSにも適用できる。

【0066】本発明は、密結合／疎結合マルチプロセッサシステム大型計算機のソフトウェアシステムを介して実現することも、また、各処理部のために専用プロセッサが用意された密結合／疎結合複合プロセッサシステムを介して実現することも可能である。また、単一プロセッサシステムでも、各処理手順のために並列なプロセスを割当てていれば、適用可能である。

【0067】本実施例によれば、各ノードで実行するデータベース演算に対応して各ノード数を決定し、また、データの分割にバラツキが存在する場合、各ノードヘデ

ータを均等に分割させ、各ノードで実行する各データベース演算をパラメタ化し、期待する処理時間均等化させるので、各ノード間で処理時間の偏りがなく、円滑にパイプライン動作させることが可能となり、高速な問合せ処理が実現可能となる。

【0068】

【発明の効果】各ノードで実行するデータベース演算に対応して各ノード数を決定し、各ノードヘデータを均等に分割させ、各ノードで実行する処理時間均等化させるので、各ノード間で処理時間の偏りがなく、高速な問合せ処理を実現することができる。

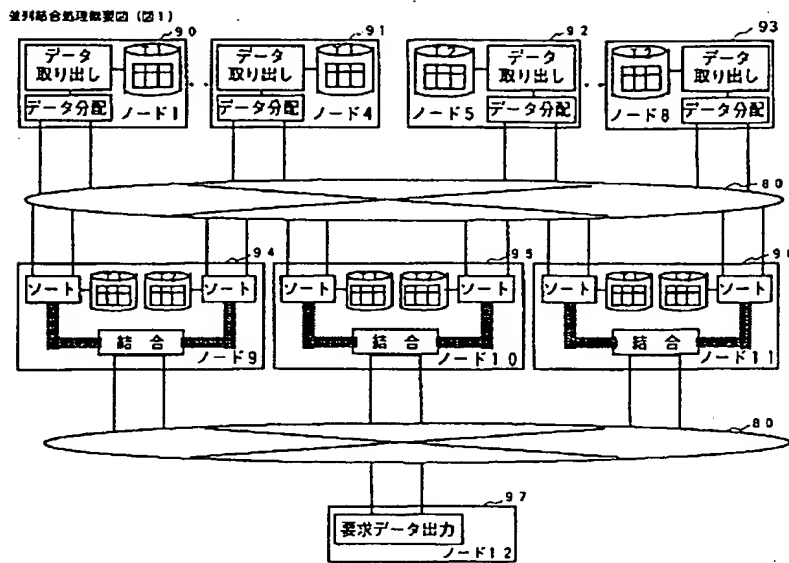
【図面の簡単な説明】

- 【図1】 並列結合処理概要図
- 【図2】 データベースシステムの構成図
- 【図3】 ハードウェア構成図
- 【図4】 並列パイプライン動作の概要図
- 【図5】 データ分配処理概要図
- 【図6】 結合ノード割当て概要図
- 【図7】 スロットソート前処理化概要図
- 【図8】 スロットソート連長チューニング概要図
- 【図9】 Nウェイマージ回数チューニング概要図
- 【図10】 データベース管理システムのフローチャート
- 【図11】 データベース管理システムのフローチャート
- 【図12】 データベース管理システムのフローチャート
- 【図13】 データベース管理システムのフローチャート
- 【図14】 データベース管理システムのフローチャート
- 【図15】 データベース管理システムのフローチャート

【符号の説明】

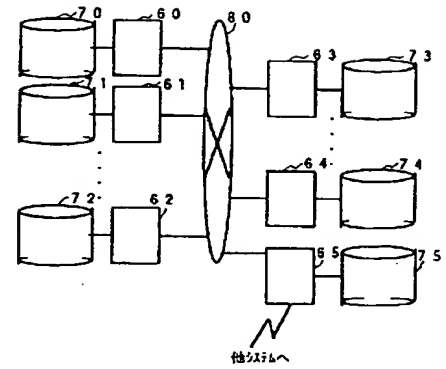
10、11…アプリケーションプログラム、20…データベース管理システム、22…論理処理部、220…問合せ解析、221…静的最適化処理、222…コード生成、223…動的最適化処理、224…コード解釈実行、30…オペレーティングシステム、40…データベース、50…ディクショナリ、80…相互結合ネットワーク、90、91、92、93、94、95、96、97…ノード。

【図1】



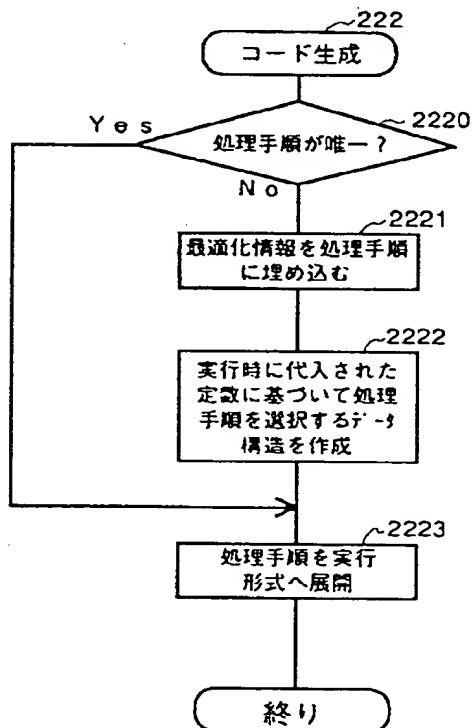
【図3】

ハードウェア構成図 (図3)



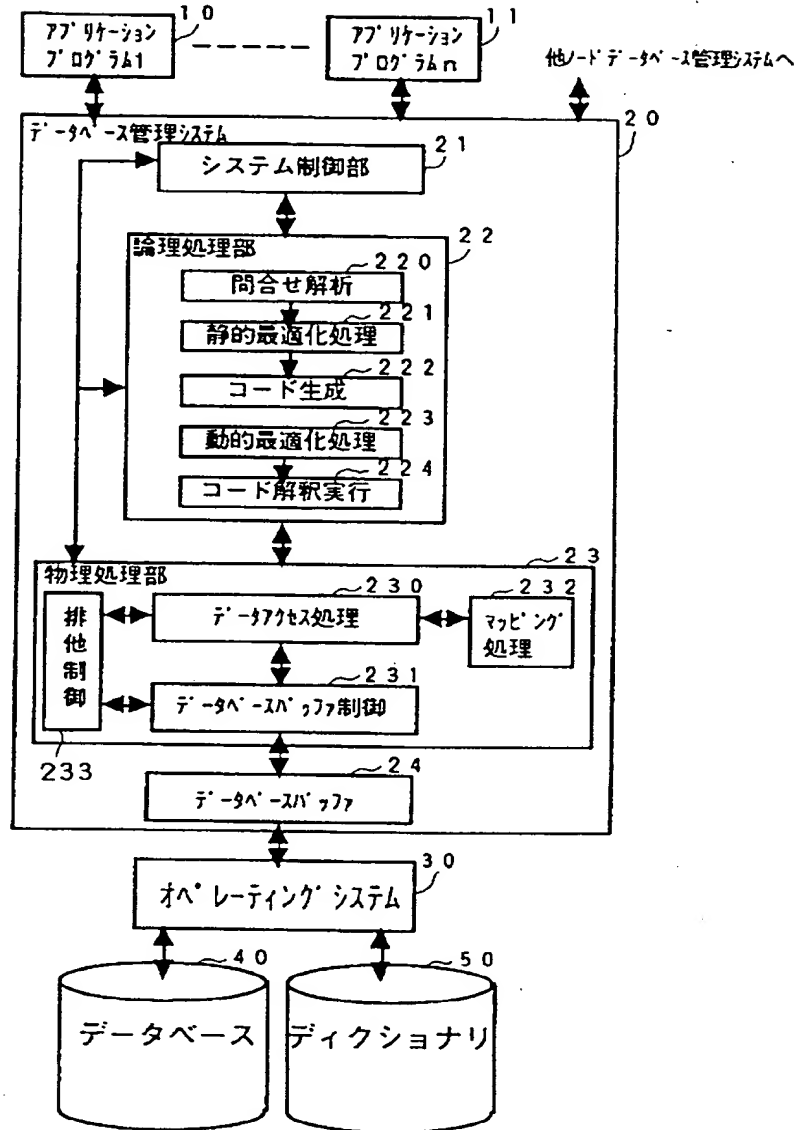
【図13】

データベース管理システムのフローチャート (図13)



【図2】

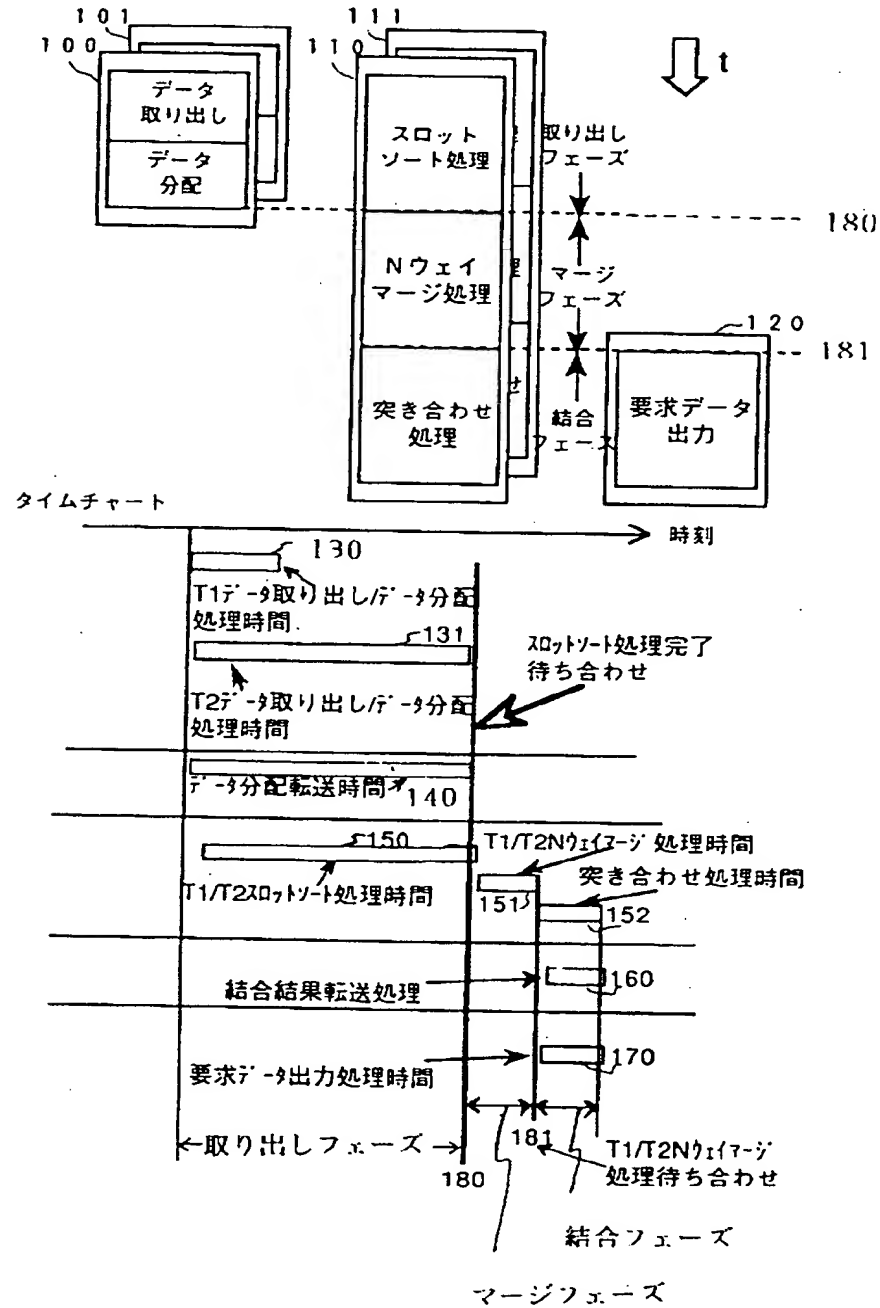
データベースシステムの構成図 (図2)



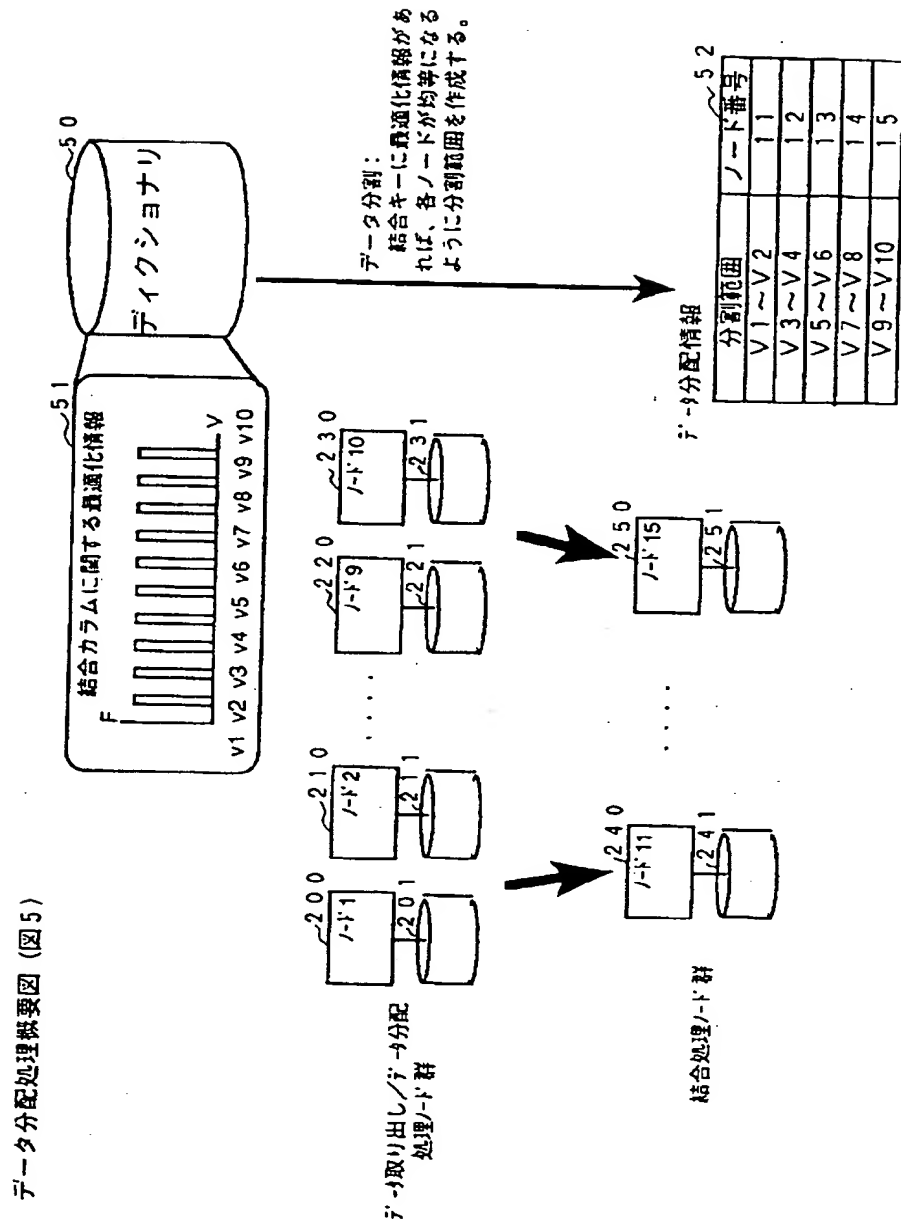


【図4】

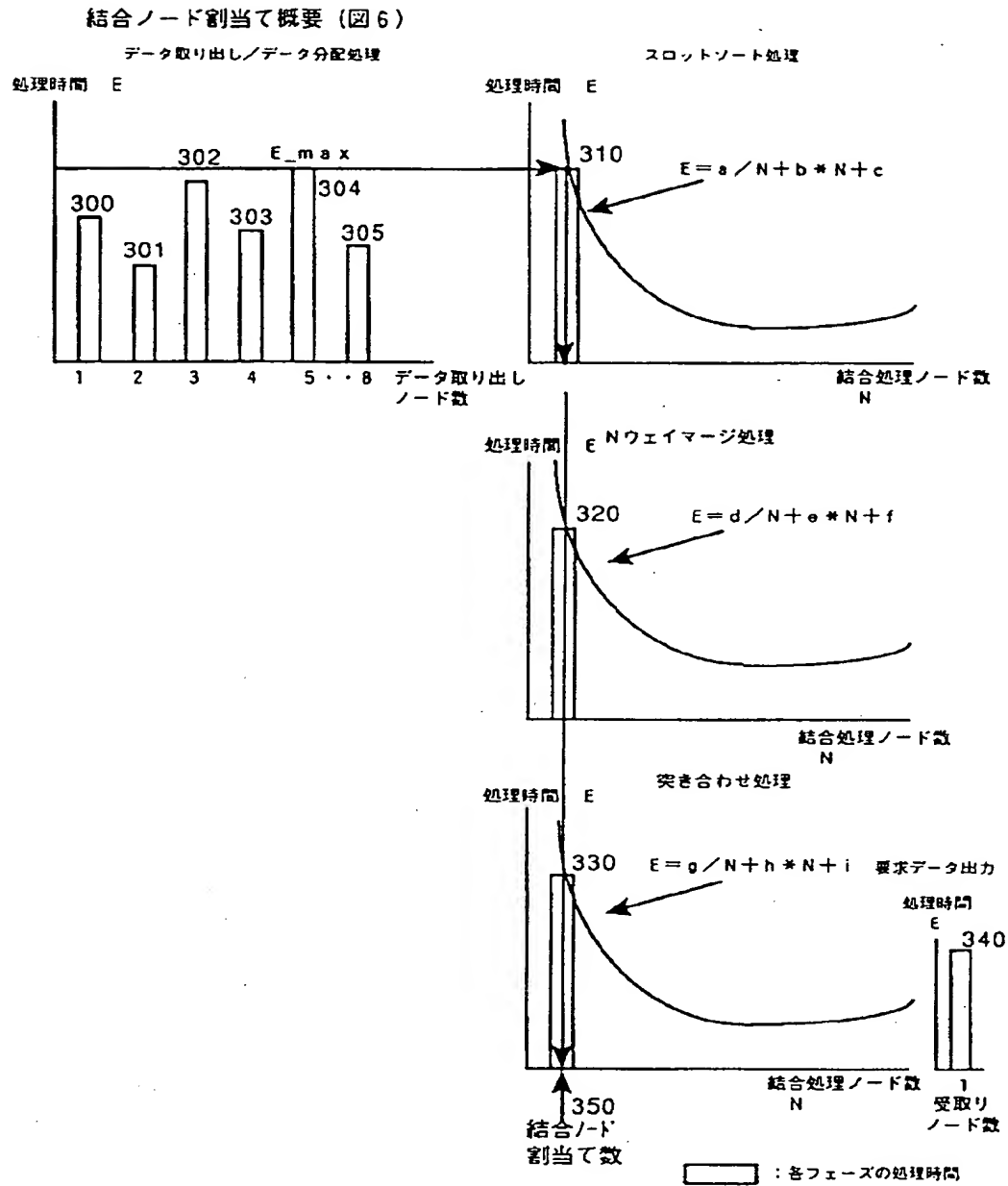
並列パイプライン動作の概要図 (図4)



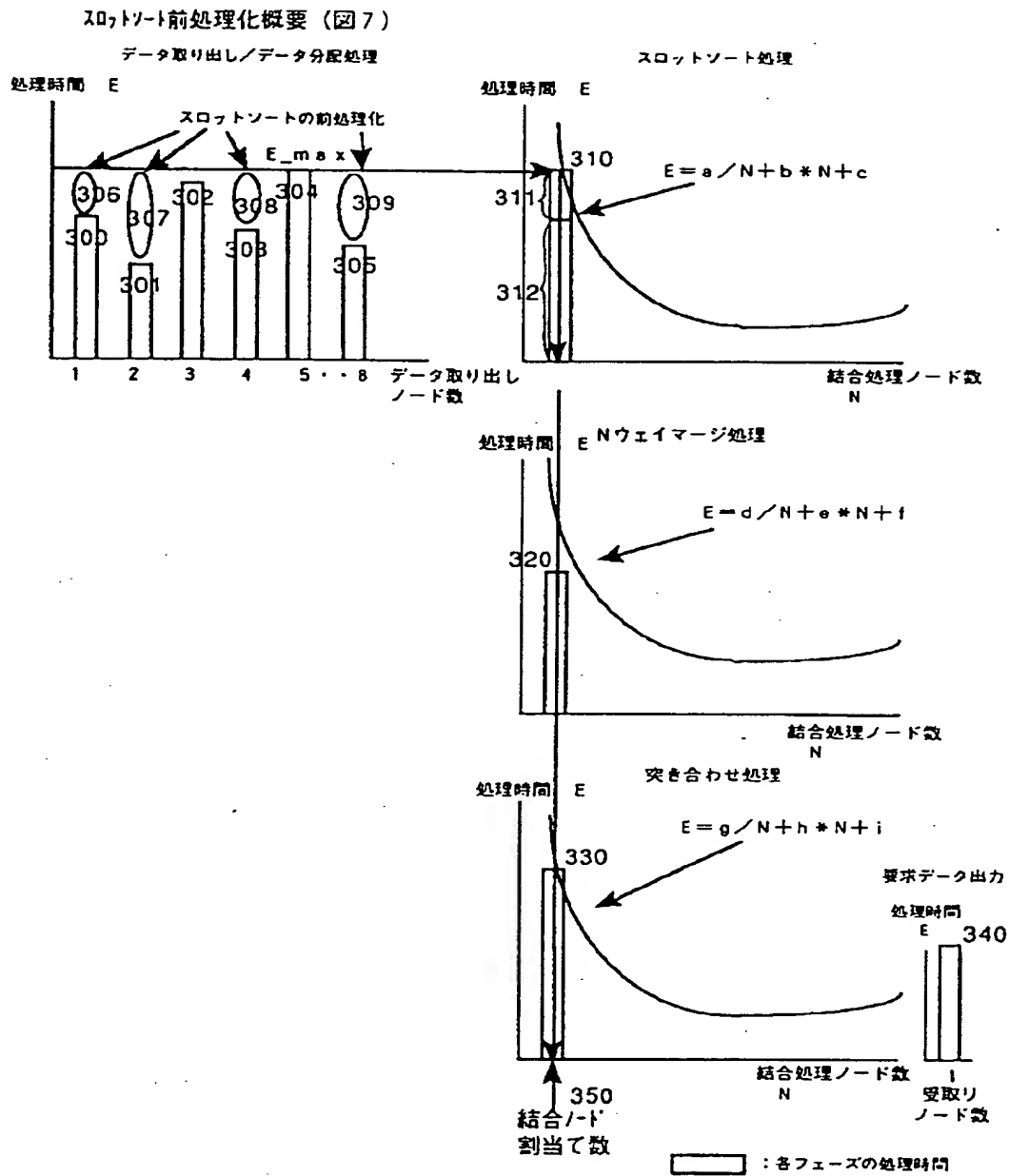
【図5】



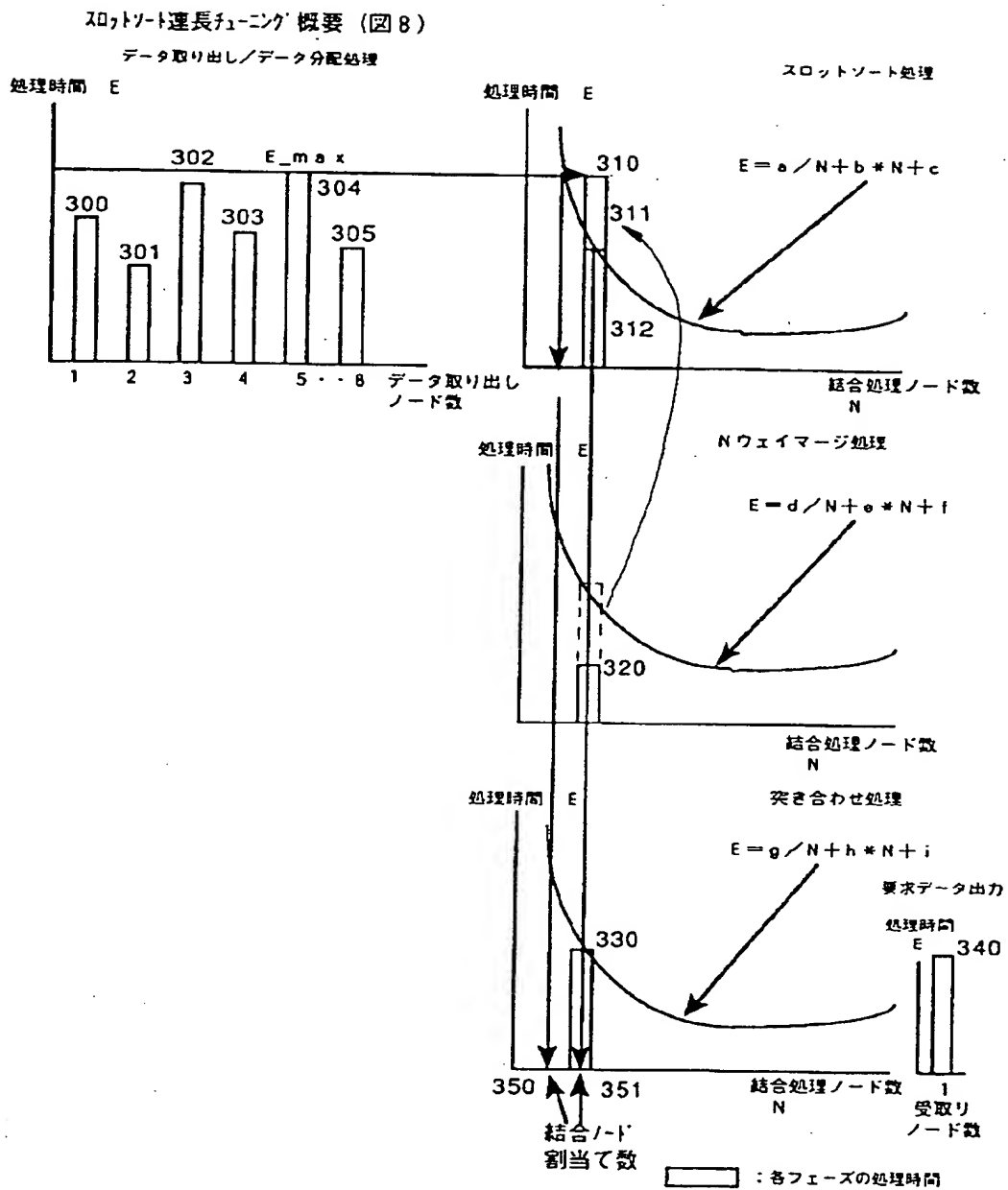
【図6】



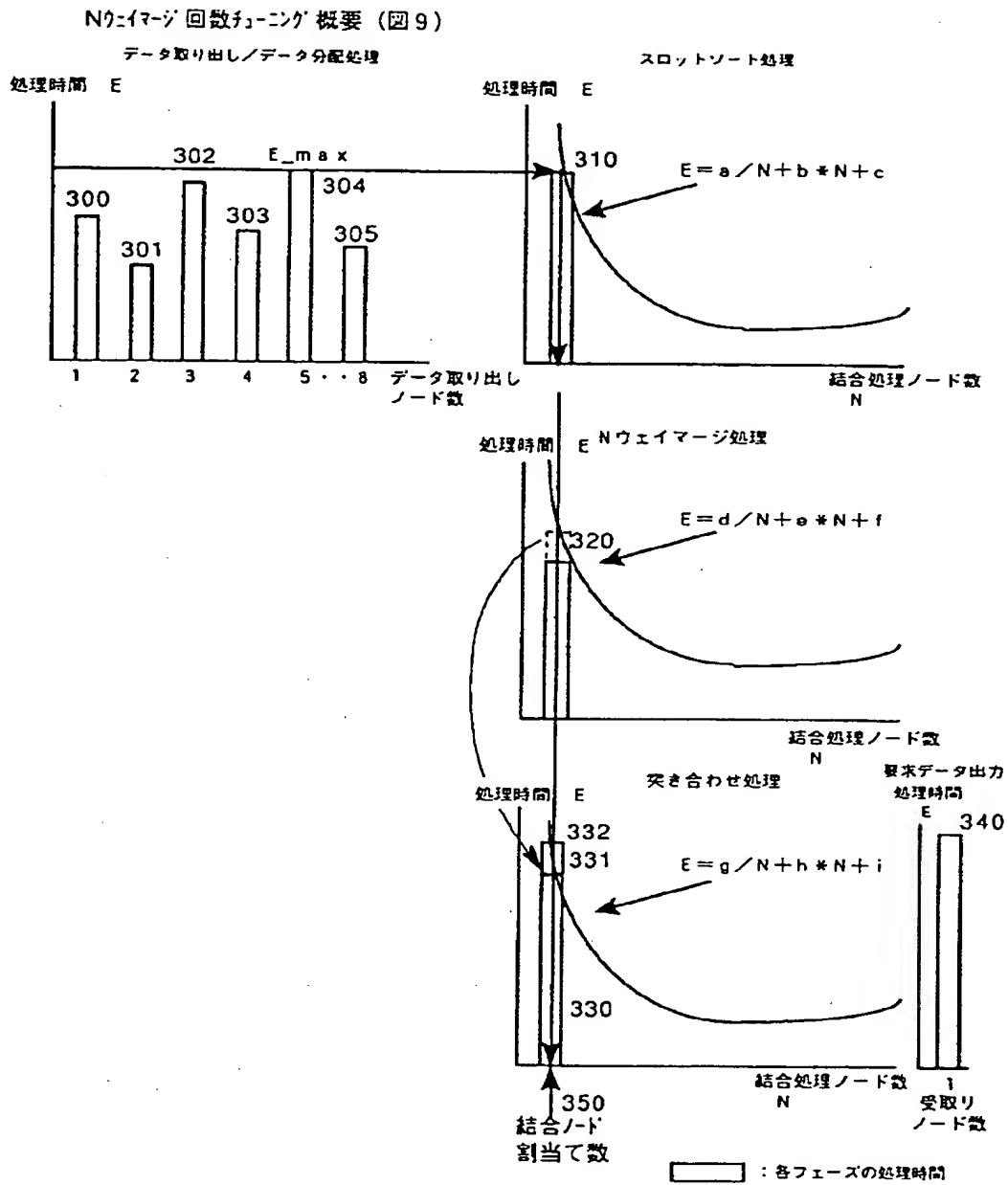
【図7】



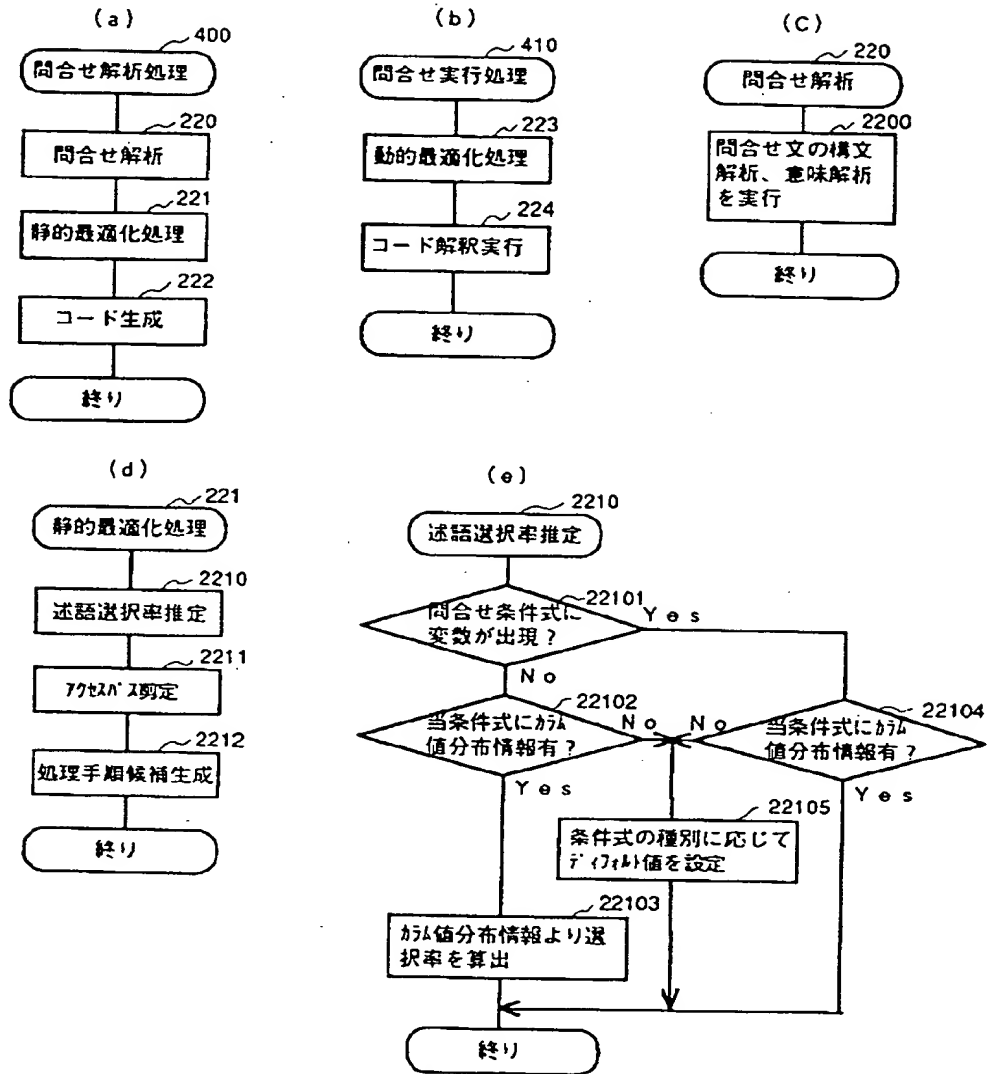
【図8】



【図9】

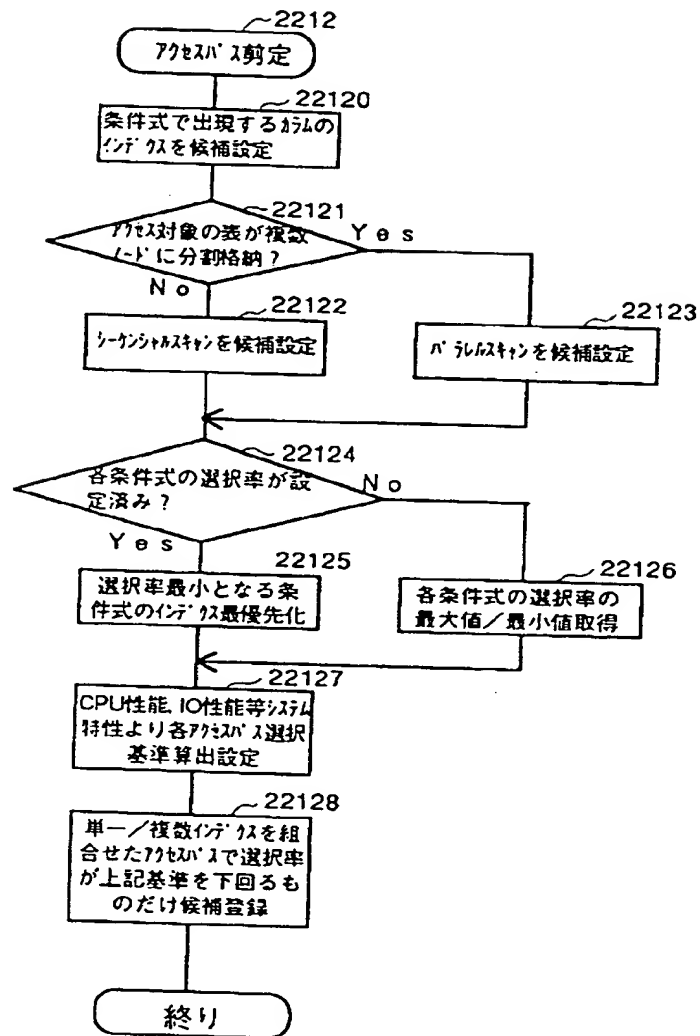


データベース管理システムのフローチャート (図10)



【図11】

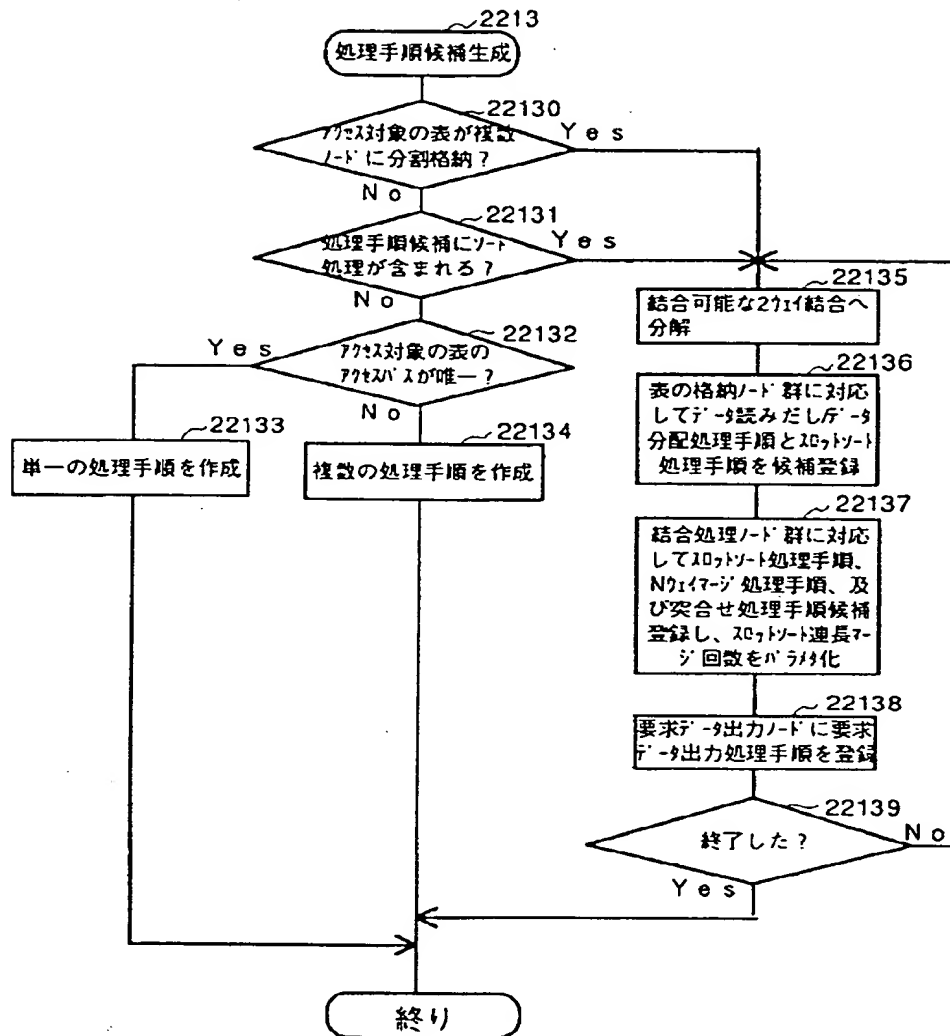
データベース管理システムのフローチャート (図11)





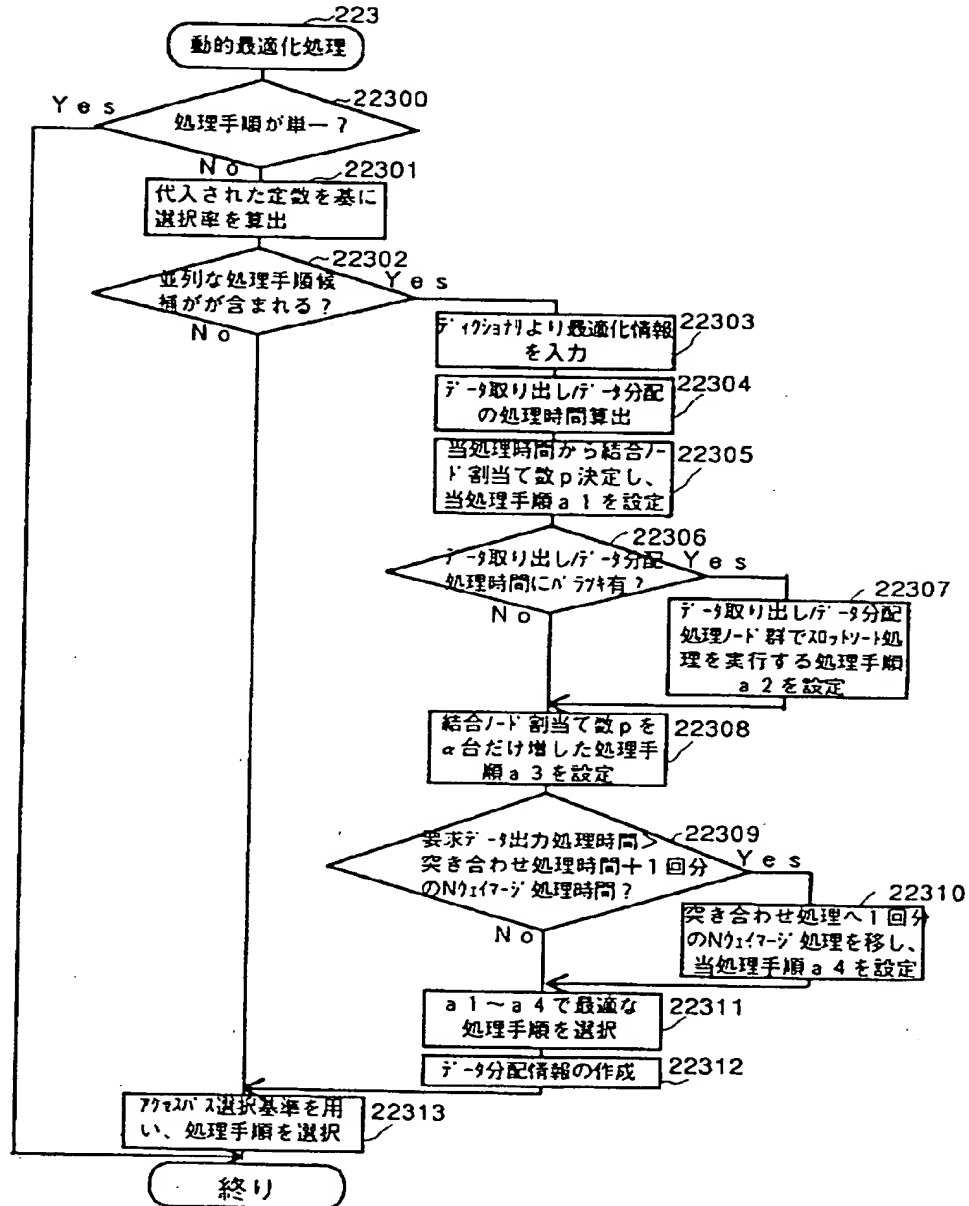
【図12】

データベース管理システムのフローチャート (図12)



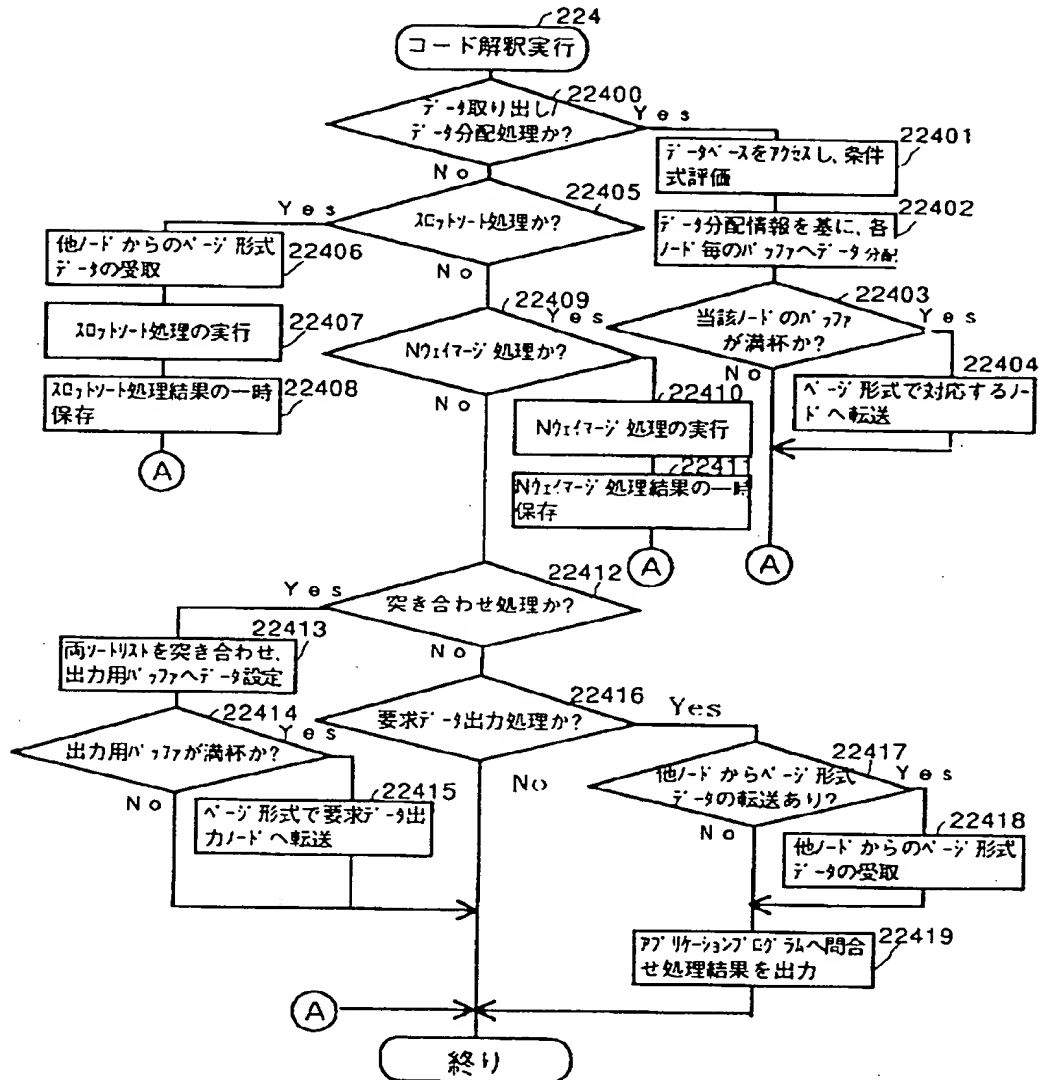
【図14】

データベース管理システムのフローチャート（図14）



【図15】

データベース管理システムのフローチャート（図15）



フロントページの続き

(72)発明者 根岸 和義  
 神奈川県川崎市麻生区王禅寺1099 株式会社  
 日立製作所システム開発研究所内

(72)発明者 鳥居 俊一  
 神奈川県川崎市麻生区王禅寺1099 株式会社  
 日立製作所システム開発研究所内

【公報種別】特許法第17条の2の規定による補正の掲載  
 【部門区分】第6部門第3区分  
 【発行日】平成9年(1997)5月20日

【公開番号】特開平6-214843  
 【公開日】平成6年(1994)8月5日  
 【年通号数】公開特許公報6-2149  
 【出願番号】特願平5-7804  
 【国際特許分類第6版】

G06F 12/00 513  
 512

【FI】

G06F 12/00 513 J 7623-5B  
 512 7623-5B

【手続補正書】

【提出日】平成8年7月15日

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】ネットワークを介して相互に接続される複数のノードを備えるデータベース管理システムであつて、

データベースを構成するデータを分散して格納する記憶手段を備え、指示されたときに、前記記憶手段に格納されたデータを取り出し、取り出したデータを他のノードに分配すべく前記ネットワークに送出する複数の第1のノードと、

指示されたときに、前記ネットワークに送出されたデータを受け取り、受け取ったデータに対するデータベース演算を実行し、その結果を前記ネットワークに送出する複数の第2のノードと、

ユーザからの問い合わせを受け付けて、該問い合わせを解析し、該解析結果に基づいて、前記複数の第1のノードのうち少なくとも一つの第1のノードに対して前記データの取りだしを指示し、前記第2のノードのうち少なくとも一つの第2のノードに対して前記データベース演算の実行を指示し、前記ネットワーク上に送出された演算結果を受け取り、前記問い合わせに対する結果を出力する第3のノードとを有することを特徴とするデータベース管理システム。

【請求項2】請求項1において、前記第3のノードは、受け付けた問い合わせの解析結果に基づいて、前記データの取りだしを指示する第1のノードを決定し、当該第1のノードにおける予想される処理時間を算出し、該処理時間に基づいて前記データベース演算の実行を指示する第2のノードを決定することを特徴とするデータベ-

ス管理システム。

【請求項3】請求項2において、前記第3のノードは、前記決定された第1のノードにおける、予想される取り出しデータ量に基づいて、前記決定された第2のノードへの前記取り出しデータの分配を当該決定された第2のノードに均等に割当てるようにすることを特徴とするデータベース管理システム。

【請求項4】請求項3において、前記第3のノードは、前記決定された第2のノードに取り出しデータを均等に割当てするための、前記各ノードの記憶手段のデータに関する最適化情報を記憶している記憶手段を備えることを特徴とするデータベース管理システム。

【請求項5】請求項3において、前記第3のノードは、前記決定された第2のノードに取り出しデータを均等に割当てのためにあらかじめ定められたハッシュ関数を利用することを特徴とするデータベース管理システム。

【請求項6】請求項2において、前記複数のノードは、それぞれ独立に処理を行い、前記決定された第2のノードは、前記決定された第1のノードからの分配されたデータを逐次入力し、入力されたデータごとに処理を行うことを特徴とするデータベース管理システム。

【請求項7】請求項6において、前記複数の第1のノードの各々は、該第1のノードで分配するデータをさらに並び替えることを特徴とするデータベース管理システム。

【請求項8】請求項7において、前記第3のノードは、前記決定された第1のノードにおける予想される処理時間の算出結果から、より先に処理が終了する第1のノードに対して分配処理後に当該第1のノードにおいて並び替えをするように決定することを特徴とするデータベース管理システム。

【請求項9】請求項8において、前記第3のノードは、前記処理時間に基づいて決定した前記第2のノードの台

数を、所定数増加させるように決定することを特徴とするデータベース管理システム。

【請求項10】請求項9において、前記決定された第2のノードは、前記第1のノードにおける分配されたデータの並び替えの終了後にマージ処理をすることを特徴とするデータベース管理システム。

【請求項11】請求項7において、前記第2のノードは、前記第1のノードにおける分配されたデータの並び替えの終了後にマージ処理を行い、該マージされたデータに基づいて問い合わせに対する突き合わせの処理を行うことを特徴とするデータベース管理システム。

【請求項12】請求項11において、前記第3のノードは、前記突き合わせの処理および前記問い合わせに対する結果の出力の処理における予想される処理時間を算出し、該算出結果に基づいて、前記出力の処理の処理時間が、前記突き合わせの処理の時間より大きい場合には、前記突き合わせの処理の中で前記マージ処理を行わせることを特徴とするデータベース管理システム。

【請求項13】情報を記憶する記憶手段を備え、前記記憶手段から情報を取り出して他のノードに取り出した情報を分配し、並び替えを実行する複数の第1のノードと、

該第1のノードから分配された情報を並び替え、該並び替えられた情報が複数ある場合にはそれらをマージし、該マージされた情報に基づいて問い合わせに対するマージと突き合わせとを実行する複数の第2のノードと、前記問い合わせを受け付けて、該問い合わせを解析して問い合わせの処理手順を作成し、該問い合わせの解析結果に基づいて、実行処理を行う第1のノードおよび第2のノードを決定する第3のノードとを備えるデータベース管理システムにおける問い合わせ処理方法であって、前記第3のノードは、前記問い合わせの解析結果に基づいて、前記情報の分配と前記並び替えとを実行する第1のノードを決定し、当該第1のノードにおける予想される処理時間を算出し、該処理時間に基づいて、前記マージと前記突き合わせとを実行する第2のノードを決定し、前記決定された第1のノードのそれぞれは、前記問い合わせの解析結果に基づいて前記記憶手段から情報を取り出して該取り出した情報を他のノードに分配し、並び替えを行い、前記決定された第2のノードのそれぞれは、前記第1のノードから分配された情報を並び替え、該並び替えられた情報が複数ある場合にはそれらをマージし、該マージされた情報に基づいて問い合わせに対するマージと突き合わせとをし、前記第2のノードから得られた、問い合わせに対する結果を出力することを特徴とする問合せの処理方法。

【請求項14】ネットワークを介して相互に接続される複数のノードを備えるデータベース管理システムであって、

データベースを構成するデータを分散して格納する記憶手段を備え、指示されたときに、前記記憶手段に格納されたデータを取り出し、取り出したデータを他のノードに分配すべく前記ネットワークに送出すると共に、他のノードから前記ネットワーク上に送出されたデータを受け取り、受け取ったデータに対するデータベース演算を実行し、その結果を前記ネットワークに送出する複数の第1のノードと、

ユーザからの問い合わせを受け付けて、該問い合わせを解析し、該解析結果に基づいて、前記複数の第1のノードのうち少なくとも一つの第1のノードに対して前記データの取りだしの指示、および、前記データベース演算の実行の指示を行い、前記ネットワーク上に送出された演算結果を受け取り、前記問い合わせに対する結果を出力する第2のノードとを有することを特徴とするデータベース管理システム。

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0012

【補正方法】変更

【補正内容】

【0012】

【課題を解決するための手段】本発明は、上記課題を解決するために、ネットワークを介して相互に接続される複数のノードを備えるデータベース管理システムであって、データベースを構成するデータを分散して格納する記憶手段を備え、指示されたときに、前記記憶手段に格納されたデータを取り出し、取り出したデータを他のノードに分配すべく前記ネットワークに送出する複数の第1のノードと、指示されたときに、前記ネットワークに送出されたデータを受け取り、受け取ったデータに対するデータベース演算を実行し、その結果を前記ネットワークに送出する複数の第2のノードと、ユーザからの問い合わせを受け付けて、該問い合わせを解析し、該解析結果に基づいて、前記複数の第1のノードのうち少なくとも一つの第1のノードに対して前記データの取りだしを指示し、前記第2のノードのうち少なくとも一つの第2のノードに対して前記データベース演算の実行を指示し、前記ネットワーク上に送出された演算結果を受け取り、前記問い合わせに対する結果を出力する第3のノードとを有する。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0013

【補正方法】変更

【補正内容】

【0013】前記第3のノードは、受け付けた問い合わせの解析結果に基づいて、前記データの取りだしを指示する第1のノードを決定し、当該第1のノードにおける予想される処理時間を算出し、該処理時間に基づいて前

記データベース演算の実行を指示する第2のノードを決定することができる。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0014

【補正方法】変更

【補正内容】

【0014】前記第3のノードは、前記決定された第1のノードにおける、予想される取り出しデータ量に基づいて、前記決定された第2のノードへの前記取り出しデータの分配を当該決定された第2のノードに均等に割当てるようにする。

【手続補正5】

【補正対象書類名】明細書

【補正対象項目名】0015

【補正方法】変更

【補正内容】

【0015】前記第3のノードは、前記決定された第2のノードに取り出しデータを均等に割当てるための、前記各ノードの記憶手段のデータに関する最適化情報を記憶している記憶手段を備えることができる。

【手続補正6】

【補正対象書類名】明細書

【補正対象項目名】0016

【補正方法】変更

【補正内容】

【0016】前記第3のノードは、前記決定された第2のノードに取り出しデータを均等に割当てるためにあらかじめ定められたハッシュ関数を利用する。

【手続補正7】

【補正対象書類名】明細書

【補正対象項目名】0017

【補正方法】変更

【補正内容】

【0017】また、前記複数のノードは、それぞれ独立に処理を行い、前記決定された第2のノードは、前記決定された第1のノードからの分配されたデータを逐次入力し、入力されたデータごとに処理を行う。

【手続補正8】

【補正対象書類名】明細書

【補正対象項目名】0018

【補正方法】変更

【補正内容】

【0018】さらに、前記複数の第1のノードの各々は、該第1のノードで分配するデータをさらに並び替えるようにしてもよい。

【手続補正9】

【補正対象書類名】明細書

【補正対象項目名】0019

【補正方法】変更

【補正内容】

【0019】前記第3のノードは、前記決定された第1のノードにおける予想される処理時間の算出結果から、より先に処理が終了する第1のノードに対して分配処理後に当該第1のノードにおいて並び替えをするように決定することができる。

【手続補正10】

【補正対象書類名】明細書

【補正対象項目名】0020

【補正方法】変更

【補正内容】

【0020】前記第3のノードは、前記処理時間に基づいて決定した前記第2のノードの台数を、所定数増加させるように決定する。

【手続補正11】

【補正対象書類名】明細書

【補正対象項目名】0021

【補正方法】変更

【補正内容】

【0021】前記決定された第2のノードは、前記第1のノードにおける分配されたデータの並び替えの終了後にマージ処理をするようにしてもよい。前記第2のノードは、前記第1のノードにおける分配されたデータの並び替えの終了後にマージ処理を行い、該マージされたデータに基づいて問い合わせに対する突き合わせの処理を行う。

【手続補正12】

【補正対象書類名】明細書

【補正対象項目名】0022

【補正方法】変更

【補正内容】

【0022】前記第3のノードは、前記突き合わせの処理および前記問い合わせに対する結果の出力の処理における予想される処理時間を算出し、該算出結果に基づいて、前記出力の処理の処理時間が、前記突き合わせの処理処理の時間より大きい場合には、前記突き合わせの処理の中で前記マージ処理を行わせる。

【手続補正13】

【補正対象書類名】明細書

【補正対象項目名】0023

【補正方法】変更

【補正内容】

【0023】

【作用】第3のノードは、ユーザからの問い合わせを受け付けて、該問い合わせを解析し、該解析結果に基づいて、前記複数の第1のノードのうち少なくとも一つの第1のノードに対して前記データの取りだしを指示し、前記第2のノードのうち少なくとも一つの第2のノードに対して前記データベース演算の実行を指示する。第3のノードは、前記決定された第1のノードにおける、予想

される取り出しデータ量に基づいて、前記決定された第2のノードへの前記取り出しデータの分配を当該決定された第2のノードに均等に割当てるようにする。

【手続補正14】

【補正対象書類名】明細書

【補正対象項目名】0024

【補正方法】変更

【補正内容】

【0024】決定された第1のノードの各々は、指示されたときに、前記記憶手段に格納されたデータを取り出し、取り出したデータを他のノードに分配すべく前記ネットワークに送出する。第1のノード及び第2のノードは、それぞれ独立に処理を行い、決定された第2のノードは、前記決定された第1のノードからの分配されたデータを逐次入力し、入力されたデータごとに処理を行う。決定された第2のノードの各々は、指示されたときに、前記ネットワークに送出されたデータを受け取り、受け取ったデータに対するデータベース演算を実行し、その結果を前記ネットワークに送出する。第3のノードは、前記ネットワーク上に送出された演算結果を受け取り、前記問い合わせに対する結果を出力する。

【手続補正15】

【補正対象書類名】明細書

【補正対象項目名】0025

【補正方法】変更

【補正内容】

【0025】また、第3のノードは、前記決定された第1のノードにおける予想される処理時間の算出結果から、より先に処理が終了する第1のノードに対して分配処理後に当該第1のノードにおいてデータを並び替えるように決定する。決定された第1のノードは、分配処理後にデータを並び替える。

【手続補正16】

【補正対象書類名】明細書

【補正対象項目名】0026

【補正方法】変更

【補正内容】

【0026】さらに、第3のノードは、前記突き合わせの処理および前記問い合わせに対する結果の出力の処理における予想される処理時間を算出し、該算出結果に基づいて、前記出力の処理の処理時間が、前記突き合わせの処理処理の時間より大きい場合には、前記突き合わせの処理の中で前記マージ処理を行わせる。

【手続補正17】

【補正対象書類名】明細書

【補正対象項目名】0027

【補正方法】変更

【補正内容】

【0027】また、処理時間が予め定まっているときに、該処理時間以内で処理をさせるために第3のノード

は、前記処理時間に基づいて決定した前記第2のノードの台数を、所定数増加させるように決定する。これにより、第2のノードの台数が増加し、並び替えの処理をより短時間で処理することができる。並び替え処理の終了後にはマージ処理をすることができる。

【手続補正18】

【補正対象書類名】明細書

【補正対象項目名】0028

【補正方法】変更

【補正内容】

【0028】本発明によれば、データを取りだして分配するノードとデータベース演算を実行するノードとを決定できる。また、データの分割にバラツキが存在する場合、各ノードへデータを均等に分割させ、各ノードで実行する各データベース演算をパラメタ化し、期待する処理時間を均等化させるので、各ノード間で処理時間の偏りがなく、円滑にパイプライン動作をさせることが可能である。

【手続補正19】

【補正対象書類名】明細書

【補正対象項目名】0036

【補正方法】変更

【補正内容】

【0036】つぎに、上記各ノードの処理時間の関係について図4を参照して説明する。図4は、並列パイプライン動作を説明するための概要図を示す。図4において、100および101は、図1におけるノード1(90)からノード4(91)と、ノード5(92)からノード8(93)とにおける処理に対応し、データ取り出し処理およびデータ配分処理を実行する。110および111は、ノード9(94)からノード11(96)における処理に対応し、スロットソート処理、Nウェイマージ処理、突き合わせ処理が実行される。120は、ノード12(97)における処理に対応し、要求データ出力処理が実行される。時間軸に沿えば、データ取り出し処理およびデータ配分処理100および101で処理されたデータは、逐次スロットソート処理110および111に移り、パイプライン的に実行される。データ取り出し処理からスロットソート処理までを取り出しフェーズと呼ぶ。また、Nウェイマージ処理110および111は、それぞれのノードで単に並列に実行される。このNウェイマージ処理期間をマージフェーズと呼ぶ。さらに、突き合わせ処理の結果は要求データ出力処理120に逐次転送されてパイプライン的に実行される。この突き合わせから要求データ出力までを結合フェーズと呼ぶ。図4に示すタイムチャートは、図1に示す問合せ例を適用した場合の処理内容である。取り出しフェーズにおいて、ノード1(90)からノード4(91)における処理時間は、T1データ取り出し/データ配分処理時間130として示す。また、ノード5(92)からノード

ド 8 (93) における処理時間は、T2 データ取り出し / データ分配処理時間 131 で示し、相互結合ネットワーク 80 における転送時間をデータ分配転送時間 140 で示し、ノード 9 (94) からノード 11 (96) における処理時間は、T1 / T2 スロットソート処理時間 150 に示すように、各々実行される。図 4 において、取り出しフェーズは、スロットソート処理完了待ち合わせ 180 の時点までで終了する。また、マージフェーズにおいて、ノード 9 (94) からノード 11 (96) にお

ける処理時間は、T1 / T2 Nウェイマージ処理時間 151 に示す時間において実行される。このマージフェーズは、T1 / T2 Nウェイマージ処理待ち合わせ 181 までで終了する。結合フェーズは、ノード 9 (94) からノード 11 (96) における処理時間は、突き合わせ処理時間 152 で示し、相互結合ネットワーク 80 における処理時間は、結合結果転送時間 160 で示し、ノード 12 (97) における処理時間は、要求データ出力処理時間 170 で示し、各々その時間内に実行される。